QL Today

Volume 17
Issue 4
June ... Oct.
2013

ISSN 1432-5454

The Final Issue!

Time to say Goodbye...

The Final
DVD

All Issues +
QPC2 V4

www.QLToday.com

Contents

Advertisers

in alphabetical order

QL *Today*

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software
Kaiser-Wilhelm-Str. 302
47169 Duisburg
Germany

Tel. +49 203 502011
Fax +49 203 502012
email: smsq@j-m-s.com
email: QLToday@j-m-s.com

Editor:

Geoff Wicks
Flat 5b
Wordsworth Avenue
Derby DE24 9HQ
United Kingdom

Tel. +44 1332 271366
email: gtwicks@btinternet.com
email: QLToday@j-m-s.com

Co-Editor & UK Office:

Bruce Nicholls
38 Derham Gardens
Upminster
Essex RM14 3HA
United Kingdom

Tel +44 20 71930539
Fax +44 870 0568755
email: qltoday@q-v-d.demon.co.uk
email: QLToday@j-m-s.com

QL *Today* is published four times a year, our volume begins on beginning of June. Please contact the German or English office for current subscription rates or visit our homepage www.QLTODAY.com.

We welcome your comments, suggestions and articles. YOU make **QL *Today*** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5' disk (DD or HD) or sent via Email. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

QL *Today* reserves the right to publish or not publish any material submitted. Under no circumstances will **QL *Today*** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL *Today***. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

If you need more information about the UNZIP program which is used by our BOOT program to unpack the files, we suggest that you visit Dilwyn Jones' web site where you find more information about lots of interesting QDOS software and INFOZIP at http://www.dilwyn.uk6.net/arch/index.html

Editorial

by Geoff Wicks

The last issue of a volume of QL Today is when I traditionally look back over the previous four issues and then look to the future. This time there is no future as this is the last issue of the last volume of QL Today.

The closure of QL Today seems to have caught many people by surprise, but it should not have done. Three years ago I expressed my concern about the lack of editorial material:
'Good though our regular writers are, they cannot fill all the pages.'

A year later I wrote:
'Of some concern is a fall in the number of contributors. Thirteen writers compared with twenty one in the previous year and eighteen in the year before that.'

In the present volume we have had just 11.

The Quanta Magazine is also in trouble editorially. In the last complete volume three of the six issues had just three contributors each and a further two just four. Two of those contributors were the Editor and the News Editor. Between them they filled 24 of the 39 editorial pages in the last issue.

I have a great respect for the editor of the Quanta Magazine, who has a more difficult job than I have had as editor of QL Today. Not only does he have to produce 6 issues a year instead of 4, he is also responsible for both editorial content and production. The Quanta Magazine editor has by far the hardest job on the Quanta committee.

Quanta members have much to learn from the closure of QL Today, because, frankly, they have been disloyal to their own magazine over many years. Between 2004 and 2011 no one was prepared to be its editor, and an, already overworked, officer had to take on the task. When a new editor was eventually found it was someone who had just returned to the QL community and had just rejoined Quanta. If the Quanta Magazine is to survive the editor needs more support than he is currently getting.

Some readers may be unhappy with the negative tone of a final editorial, but perhaps there is a lesson not just for Quanta members but for us all. There is no magic formula to save the QL, but instead just a lot of hard work.

Seven years has been a long time to edit a magazine, but it has been an interesting and challenging task. To our writers who have remained loyal over such a long period our grateful thanks. Without you the magazine would not have survived against all odds to be the longest lived QL magazine with the exception of the Quanta Magazine.

Much has changed in those seven years. In 2005 the UK had 5 active traders. Now there is just one. Quanta membership has halved and can now manage only one show a year - always in Manchester. QL news is slowly drying up. It has become all too easy to build up a negative picture of the QL's future.

But for all that the QL has shown a remarkable resilience. This year we have seen a major new QL emulator and updated versions of SMSQ/E and QPC2. In turn these brought out the best in the QL-users email group and the QL Forum. About a half of Quanta members now receive their magazine electronically. And Quanta, for the first time, has had electronic participation in both committee meetings and the AGM.

There will still be life after QL Today. It is for QL-ers to decide just what that life will be. We can sit and wait for something to happen, or for others to do the work, but the best way of ensuring the QL's survival is for all of us to remain active.

A Look at the Past (and a Glimpse of the Future?)

by Geoff Wicks

When QL Today was first published in 1996 the chances of its long term survival were very low. Within a space of about two years three QL magazines had failed, and one of them, QL World, had been in trouble for about four years before that. In 1992 it had had three different publishers in the space of one year.

Originally QL World was published by Focus Magazines until the firm went into administration in 1990. It was then published by the Maxwell organisation, but this came to an abrupt end when the owner, Robert Maxwell, disappeared from his yacht in mysterious circumstances and was later found drowned. It transpired that he had been defrauding the company's pension fund, thus fuelling speculation about whether his death was an accident, suicide or murder.

QL World was taken over by a company called HHL who after one issue passed it on to Arcwind, a company whose main interest was wind surfing.

It is never pleasant to be in at the demise of a QL magazine and I have experienced the death of four.

I was aware that QL World was in serious trouble long before most of its readers. In 1992 I had written a series of 5 articles on desk top publishing and they were printed under three different proprietors. Maxwell and HHL paid promptly, but I had to wait a long time for payment by Arcwind.

After their publication Freddy Vaccha suggested I wrote a similar series on word processing. It was an interesting suggestion and there followed a difficult tripartite negotiation between Freddy, editor Helen Armstrong and myself to ensure there was editorial independence. After I had written the articles I received a confidential approach from Helen indicating that QL World was in financial trouble and asking if I was prepared to waive a fee.

Just before the articles were due to be published QL World finally closed down. Seven years later I wrote a crossword competition for Quanta with the clue "RIP 1994. Run down by wind surfers (7)".

In some ways the demise of QL World was good for the QL. We were no longer dependent on the commercial world and QL publications became a cottage industry run by QL enthusiasts. Unfortunately, at first, the cottage industry did not flourish.

There were two possible contenders to replace QL World - in the UK QReview and in America IQLR. My word processing articles were published in QReview.

Eventually QReview was taken over by IQLR but the latter collapsed in 1996 when the publisher had a serious heart attack. The future could not have looked bleaker for QL publications.

In fact the situation was so serious that it was too sensitive to tell the full story at the time. The collapse of IQLR had occurred at the start of a new volume and all traders had paid for a year's advertising in advance. We never saw our money again. More seriously readers who had just paid their subscription would also have lost their money.

Two traders, Jochen Merz and Stuart Honeyball, took the initiative in starting a new magazine and taking over the IQLR subscription lists including responsibility for subscriber debts so that no reader would find himself out of pocket. They were able to secure a highly respected QL-er, Dilwyn Jones, as editor.

The first issue with a cover appropriately proclaiming "The Race Must Go On" was produced in just 6 days with most of the material being provided by the traders. It had 60 pages and 14 trader adverts and the uncertainty of the three people most closely involved in its production was clear from their editorial comments.

To add to the complications the first edition of QL Today was produced before the days of easy electronic communication and my contribution was lost in the post. Jochen and I spent a whole evening trying to get the protocol right as I tried to transmit it electronically. It was only with hindsight that I realised that the problem was that he pronounced the letter Z as "Zee" and I as "Zed".

Initially QL Today was published bimonthly and for the first 6 volumes a German language supplement was also produced. The frequency of publication from volume 10 was reduced to five issues per year with a long gap during the` summer when experience showed there was a shortage of copy. The magazine went quarterly from volume 12.

Stuart Honeyball's active participation in QL Today ended after one year and his place as UK distributor was taken over by QBranch. Although QBranch's enthusiasm for QL Today contributed much to the magazine and ensured its survival over many years, the link had to be broken in 2009 because of accumulating debts. The UK office was then taken over by Bruce Nicholls of Quo Vadis Software.

Editorially the magazine has been relatively stable. Dilwyn Jones remained as editor until 2005 when excessive demands at his day job meant he had to relinquish the post. Since then QL Today has been edited by Geoff Wicks. In 2000 Bruce Nicholls joined the magazine as Co-editor. In a sense he is the unsung hero of QL Today working in the background including as a faithful proofreader.

During the life of QL Today publishing in general has undergone a revolution. In the early days copy arrived mainly on floppy disks, but email has now become universal and most copy arrives electronically. The last time a writer sent copy to me by post, I never received it and had to ask him to resend electronically. Some months later I discovered it had been incorrectly delivered to a failed business below my flat.

The move to electronic transmission of copy was most difficult for advertisements as many traders prepared them in Line Design. The advertisements had to be printed out as hard copy and either scanned for electronic transmission or sent by post. This sometimes meant some loss in quality. In 2000 I had printer problems and could not produce quality hard copy. At short notice I tackled a problem that many said was impossible and successfully managed to transfer a LineDesign page to a PC graphics format. When I launched the new Just Words! website two years ago the article on how to do this was one of the most popular downloads.

One technical innovation that Quanta successfully achieved over 6 years ago, the production of an electronic version of the magazine, has eluded QL Today. Two problems could not be overcome without a radical change in both production and design. The software on which QL Today is produced can only produce bit map pdf files which are far too large for use in an electronic magazine. Another problem is the A4 two columnar format of QL Today which is not a good format for reading from a screen.

In 2011 Rainer Wolkwitz scanned all 15 volumes of QL Today, both English and German, and QL Today was able to give all subscribers a DVD containing an electronic archive. A disc of volumes 16 and 17 comes with this issue. As icing on the cake Brian Kemmett has kindly indexed all English issues of QL Today.

The news that this is the last issue of QL Today came as a severe blow to many readers and sparked off a lengthy discussion on the QL-users email group. There were many suggestions for alternative means of publication which, frankly, did not always look at the realities of the situation. There were two main reasons why we made the decision to close QL Today. The precipitating problem was a rise in postage charges, but in the background was also a shortage of writers.

Last year the German postal service abolished the bulk postage rates that QL Today had been using with the possible doubling or even tripling of our postage costs. Some readers suggested moving the printing of the magazine to the UK, but this would not have solved the problem. QL Today's readership is spread throughout the world and it is international postage costs that are the biggest problem.

Some idea of the costs can be gained from Quanta's experience. Members who opt for a printed magazine pay a postal supplement based on actual postage costs. The current supplement is £4 for UK members; £22.50 for European members; and £29.50 for the rest of the World.

It is difficult to make a strict comparison as QL Today is published quarterly and the Quanta Magazine bimonthly, but QL Today is a heavier publication. Nevertheless the figures give some idea of how large a subscription rise would be necessary for the magazine to remain financially viable.

The question then arises about how many subscribers would be prepared to pay a substantially increased subscription. Again we can look at the experience of Quanta, but with the proviso that it is not strictly comparable. In 2012 Quanta raised its subscription for the first time in over 20 years. This has resulted in a loss of membership of at least 15% and possibly as high as 40%. (The difference between the two figures is because of members who have continued to pay the subscription at the old rate and who are ignoring Quanta's letters asking whether they wish to remain a member.)

If a printed magazine is no longer viable, then what about an electronic magazine? There have been many suggestions about how this could be done with content management systems allowing writers to input text directly, but no one addressed the issue of where these writers would come from. In the last twelve months just 7 people have written more than once for either QL Today or the Quanta Magazine and only 4 others have contributed a single article. (These figures exclude Quanta committee members writing specifically on internal Quanta matters.) 11 writers is barely sufficient for one magazine let alone two.

Put in this way the future for QL publications appears bleak, but no bleaker than the situation when QL Today was launched in 1996. Who would have thought 17 years ago that QL Today would go on to be the longest published QL magazine with the exception of the Quanta Magazine?

If we want there to be a life after QL Today then we have to be prepared to adapt. It is no coincidence that the one UK QL trader that has survived has adopted a different trading technique from QL traditions with an emphasis on using the internet. Rich Mellor has a contact list of about 1,000 QL-ers. That is more than 5 times the current readership of QL Today and 9 to 10 times the latest membership figures for Quanta. As I argued when Rich first released his figures: *'There is a large part of the QL universe that is ripe for exploration'.*

The tool for reaching these people is the internet. Indeed the survival of the QL may come in an acknowledgement of the shortcomings of subscription based magazines and organisations. For survival QL information must become more open and available to all QL-ers.

There are at the moment a wide variety of QL websites and online discussion groups with, in particular, a comprehensive site covering most QL needs run by Dilwyn Jones. There are, however, some serious shortcomings. There is, for example, no comprehensive online news service. I run a short summary of news on the Just Words! website and in the past Quanta has had more in depth news. However, Quanta's web problems are continuing and, at the moment, it cannot be trusted to provide a consistent service to either its members or to other QL-ers.

Equally there is no provision on the internet for in depth content. Both the QL-users email group and the QL Forum successfully provide a means for QL-ers to quickly communicate with one another and for short comments and discussion, but there is nowhere for more detailed articles. One result of the closure of QL Today is that Norman Dunbar has finally had the feedback that he has previously unsuccessfully asked for. It has become clear that several readers have appreciated the assembler coverage. Equally there is a place for articles on Hardware, Games and SuperBasic.

Setting up an in depth website should not be a problem. Financially the costs would be minimal. For example, Dilwyn's site is hosted by Tony Firshman whose charges are well below those of commercial providers. I would also have space on my site.

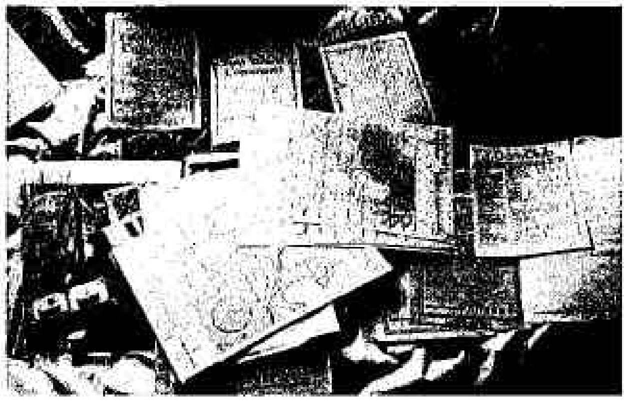
There are some snags and dangers of a site that is completely open and not subscription only. The site would have to be constantly alert for legal, copyright and potential libel problems. For these reasons there would have to be some form of editorial control making it impossible for writers to input their material directly. A decision would also need to be made on the format of the content; the length of time articles would be displayed; whether a 'house style' would be required: and whether material would be archived.

QL Today may be dead, but it does not mean that the demise of the QL itself is inevitable. The latter is a decision that QL-ers have still to take.

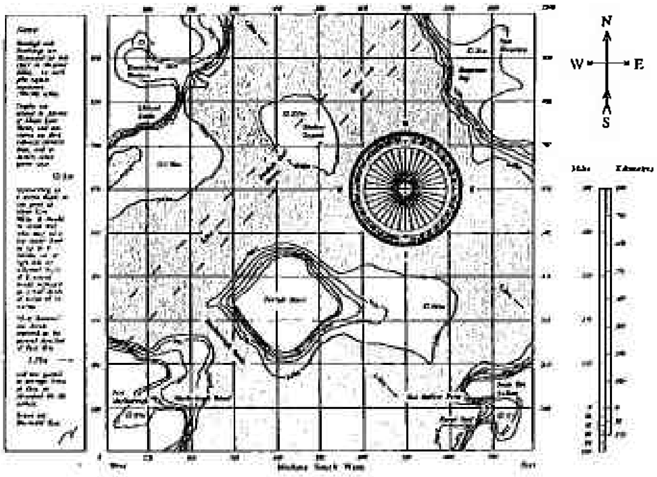
Fleet Tactical Command

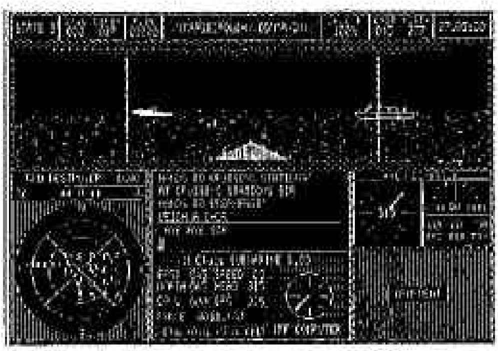
by Peter Scott

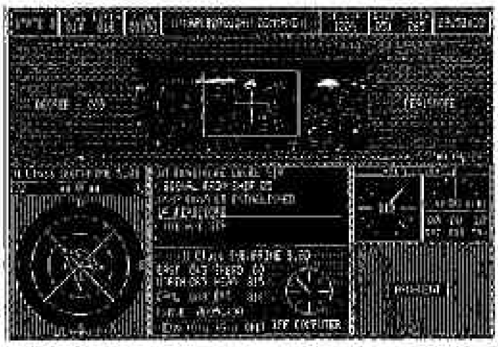
To retrogaming enthusiasts unfamiliar with the QL, the machine is light on games with few talents apart from maybe QL Chess and Matchpoint. With a good range of games covering various genres, we all know this isn't true but even then some real classics remain hidden. One of the best QL games of all Fleet Tactical Command (FTC), is nearly lost to time and totally unknown outside the QL scene. In sadly my last review for QL Today (thank you guys for all your hard work over the years!) I'll pay tribute to one of the real super heavy weight pocket battleships of the QL gaming history.



Fleet Tactical Command is a naval battle simulation with a scope, depth and eye for detail that is quite remarkable considering it was released late into the QL's commercial life in December of 1989. Development took two years alone and the game consisted over 500K of machine code listings. Early versions of the game saw two players each on a networked QL each commanding a sixteen ship fleet named either Marlbrough and San Stefano. The aim is to keep their home port stocked with supplies while hopefully preventing the other player from supplying their own, thus leaving their home port open to be captured. No mean feat over 1000 x 1000 square miles of sea and islands. The game was developed further at a steady pace offering player versus computer and even two player on the same machine.







The game itself plays out with a screen display a bit like a wireframe flight simulator. You can look all around you, use keys for controls and also type an impressive range of commands into the parser (more on this later). The ships themselves bob and sway on the sea with everything from localised currents, wave swell and changeable weather affecting your progress. Storms can blow your ships off course, damage machinery, snap anchor lines and even affect how much fuel you use on the way. To further add to the fun, your crew must be kept happy with food, water, but not being kept on high states of readiness for too long. Otherwise they can take the huff and start throwing spanners into the works quite literally! Once you multiply this by 16 ships you have quite a task on your hands.

To help with this mammoth task FTC comes with an extensive collection of sea charts, ships logs, pencils, rulers, protractors, compasses and even drawing pins. Making a gaming session looking more like a wartime operations room, maybe the box set was just short of a tobacco pipe just to finish off the experience.

One feature well worth mentioning is the damage model. Ships or submarines can suffer damage by various misfortunes including slugging it out in battle, damage from stormy weather, sabotage, collisions with other ships/docks or even just poor reliability. To repair damage commands are typed in such as LAY EMERGENCY CABLES or PUMP OFFICERS MESS. Repairing the damage can play out as a mini game in itself. If the power is off then emergency cables have to be laid, then rooms maybe have to be pumped out before they can be repaired. Other sections might have to be repaired before others and the damage even knocks out your onscreen gauges and dials. I once spent a good 20 minutes trying to save a ship that had run aground in the shallows. I had to repair and pump out various parts while waiting for the weather to worsen to see if it would improve the tide to float me off. In the end the crew got so fed up of low supplies and constantly repairing things that they ending up breaking more things faster than I could get them fixed!

I could easily spend a day listing the details in FTC. All the ships handle differently according to size and even float higher or lower in the water. Submarines have their own tactics while warships each have their own weapon systems. Some docks are protected from weather better than others, while grounded or damaged ships can be towed back to ports for repairs. Ships running low on supplies can resupplied at sea by tankers which is a tricky and skilled operation best not done in choppy weather. Survivors can even be taken on board after other ships have sunk. The ships also have their own little routines with crews coming off various watches and routines which really brings your ships to life.

I can't help wonder if FTC's impressive complexity and detailing are partly responsible for not being the success it deserved. The early versions required two QL's and a standard game could last hours if not run into double figures. A lot of time and effort is needed to really get the best from a game like this. Hours or maybe even days could be spent just trying to master all the controls and features.

The support of the game was also second to none with bug update sheets, a dedicated newsletter and continuous development. Later versions would even talk to a PC version offering cross machine play, a feature rarely found in games now.

For me the sheer scope of the game, impressive details and just for pushing the QL to the absolute limit this must rank as one of the best QL games. If only the original authors Robin J A Barker and Andrew Hopkins of Di-Ren could be tracked down and the game's re-release secured.

Glossary of Abbreviations and Terms Part 7 - T to Z

by Dilwyn Jones and Lee Privett

Task Switching Switching between programs in memory, e.g. when you have Quill and Abacus in memory, you can type into only one program at a time, but can switch back and forth between the two. This is not the same as Multi Tasking (q.v.)

TCP/IP Transmission Control Protocol/ Internet Protocol, These two are usually used together as a TCP/IP 'stack'. It's called that because TCP builds on IP (hence the actual abbreviation is 'TCP over IP'). It is called a 'stack' because IP is built to work on the actual hardware which implements internet communications. TCP is built on IP to provide advanced communications features, and possible applications are built on TCP to allow the user to use the communication capabilities, in a layer-like fashion. TCP is protocol between hosts in packet-switched computer communication networks, and in interconnected systems of such networks. The TCP part (or 'layer') handles Data Transfer, Error Detection and Correction, Flow Control, Multiplexing (emulating several communications channels over a single medium), Connecting to hosts, and Security aspects of communications. IP on the other hand is a protocol which handles how packets are distributed over packet switched networks, including addressing (where the data is sent from and where it's received), transmitting and receiving, and packing of non-packed, or stream data. Usually the IP layer handles the actual hardware which is used for Internet communication, although there might be another layer in case different hardware can be used (e.g. telephone lines, Ethernet, etc.)

THING Horrible term for a general purpose facility built into the enhanced QL systems with pointer environment etc. The designers found it hard to give this facility an accurate name due to the general nature of the beast, so they called it THING. That same generality makes it hard to describe in simple terms what a Thing is. The closest we can get is that it is a part of memory with a name of some kind. This part of memory contains a facility of some description (it may be a menu, an extension, a routine, a program and so on). The operating system maintains a list of these 'things' and a programmer can use them by looking through the list for a thing's name, and call it as required. Don't worry too much about things as a user - the QPAC2 manual says that 'things rarely go bump in the night', you can manage quite well without having to fully understand them. Anyone who used Michael Crowe's QL MegaToolkit will know that it provided a broadly similar facility called a WOTSIT. Who chooses these names, I wonder?

TIFF
Tagged Image File Format, a graphics file format supposedly to be the standard image format of the 1980's

Tilde
The ~ symbol

TK2
Abbreviation for Toolkit 2, a commonly used set of extensions to SuperBASIC providing additional 'words' to enhance the BASIC language understood by the QL. It was originally written by QL guru Tony Tebby and available as a plug in EPROM chip for the QL. Nowadays, it is commonly built into expansion cards such as the Trump Card, Gold Card and Super Gold Card, and also included with the SMSQ versions of the QL operating system. If your system does not have a copy of this (only older systems, or unexpanded systems are likely to suffer this) it is well worth getting one

TLA
Three Letter Acronym, such as BTW for By The Way

Toolkit
Software which extends the system by providing a number of extra keywords for the BASIC interpreter, or sometimes a programming package consisting of some building blocks you can use to create some new software.

Transient Program
A user program which explicitly starts and ends (apart from when it crashes of course). The QL sets aside a block of memory for programs to run in called the Transient Program Area.

Transparent Border
The QL can put a transparent border around a window by omitting the colour number in a border command. Thus BORDER #1,10 puts a 10 pixel border around the window 1, without actually colouring in the border area. On some systems, colour 128 is also transparent, so BORDER #1,3,128 would put a transparent 3 pixel border area around window #1.

Trump Card
An all singing all dancing interface card for the QL adding floppy disk interface, memory, printer buffer and toolkit 2. The Trump Card gives a QL a maximum of 896K of RAM (although earlier versions added less memory)

TT
Tony Tebby, QL designer and guru or a version of the Atari ST, for which you can get a QL emulator

TURBO
Name of a BASIC compiler program for the QL.

Turtle Graphics
Graphics drawing commands which work by a series of instructions telling how far to draw, what angle to turn before starting to draw the next line and so on. Similar in principle to how you would instruct a robotic turtle (a little robot with a pen which can draw on the paper it runs over) to draw shapes.

UB
Unsigned Byte. Describes the sound files used by the Sampled Sound System (SSS) on Q40, Q60, QPC2 and Amiga QL emulator

uQLX
A QL emulator written by Richard Zidlicky, which runs on the Linux operating system.

ULA
Uncommitted Logic Array, a type of logic chip

UNIX
UNiplexed Information and Computing Service, an operating system written by Ken Thompson of Bell Labs in the 1960s.

UNZIP
A program which decompresses compressed files produced with the ZIP program.

uQLx
A shareware QL emulator for Unix based systems. See QL Today volume 1 issue 4 for more details. The author is Richard Zidlicky

USB
Universal Serial Bus, a very fast replacement for the serial ports originally used on older computers, now in it 3rd version but maybe superseded by Lightpeak (Intel) or Thunderbolt (Apple)

Variable
A part of memory holding a value of some sort and usually referred to by a name. So LET value=6 stores the number 6 in memory and gives it the name 'value' which we can then use to specify which of all the values stored we are referring to.

Vector
Text Technique used for printing text smoothly in various sizes by storing the instructions for how to draw each character as a series of lines and curves ("vectors"). The computer can then draw nice smooth-edged characters on the screen or printer without having to magnify the characters and produce ugly blocky looking characters. A similar system can be used for graphics too. On the QL the best known example of Vector Text and Vector Graphics is the Line Design software from PROGS.

VER$
Important function in QL SuperBASIC and SBASIC which returns the characters identifying the version of SuperBASIC or SBASIC on this computer, e.g. LET a$=VER$: IF a$='JSL1' THEN PRINT"Minerva ROM installed".

VGA
Video Graphics Adaptor for the PC. On the QXL, for example, a VGA display refers to a screen mode 640 pixels wide by 480 pixels deep

Virus
Term used to refer to a malicious piece of software which sometimes attempts to damage files on your computer or steal information from your computer, or even sometimes take over control of your computer. Most viruses work on Windows, Mac and Linux computers and we are very lucky that on the QL we very rarely get our computer infected by these little nasties. The name 'virus' is used because a computer gets infected with a little nasty which affects the way it works, which is a little bit like a biological virus infecting an animal or person.

VRAM
Video Random Access Memory

WIKI
A WIKI is a website that allows users to update or add content to its database of information. For example, Rich Mellor's QL Wiki at
http://www.rwapadventures.com/ql_wiki/

WIN
The device name usually used by QL systems for a hard disk. For example, WIN1_ is usually hard disk drive number 1. The abbreviation WIN comes from what was originally called a Winchester hard disk.

Window
A defined area of the screen, identified by a channel number. For example, when you start up a QL, the red area of the screen is referred to as Window #1, the black area where you type in commands is known as window #0 and the white area where BASIC programs are LISTed is known as Window #2.

Window Manager
The Window Manager. This is part of the Pointer Environment (or Extended Environment). Provides a set of menu and display routines which a programmer can access to ensure that programs have a 'standard' appearance, or programs which look consistent with each other. Always used in conjunction with PTR_GEN (see above)

Windows
Framed glass in houses and offices allowing us to throw PCs out of. Alternatively, an operating system for PCs whose main virtue is to allow us to run QL emulators on, such as QPC2, QemuLator, QLay and QL2K.

WMAN
See Window Manager above.

WORD
Unit of computer memory. A Word is 2 bytes, or 16 bits of memory. Can store numbers from 0 up to binary 1111 1111 1111 1111

WORM
Write Once, Read.Many times. A device which can only be saved to once, and from then on only read, like a CD

WP
Word Processor. Also the term used for the USA equivalent of Quill

WWW
World Wide Web - a layer of the Internet, devised in Switzerland some time after the basic Internet came into domestic use

WYSIWYG
What You See Is What You Get, normally used when specifying how close what you see on the screen will be to how it would appear when printed on paper

WYGIWYS
What you Get Is What You See, the new way of displaying application menus maybe

XML
Extensible Markup Language, a web sub language to define documents

Yobibyte
Is a very very large amount of data

ZIP
Term used for a commonly used program to compress files into a single large archive. 'Compress' means making the files occupy less space on your computer. The term ZIP is also used for a completely separate software system used to write adventure games.

ZIP DRIVE
A storage system using a type of removable cartridge, mini-hard disk or superfloppy, made by a company called Iomega.

I hope you find this list useful. If you come across any other commonly used QL abbreviation, let me know and I'll add it to the list and update it from time to time.

Programming in Assembler, Part 34 The End of an Era, or is it?

by Norman Dunbar

Introduction

In the last issue, we ended up with a LibGen application that was getting somewhere. But it's not finished yet. This issue might well be the last paper copy of QL Today that you receive, but I have no current plans to stop development of this utility, nor to stop writing down stuff as I go along!

I need to take a slight diversion into creating dynamic application sub-window menus before I can finish the utility properly. Unfortunately, this issue will not be continuing the program's development as I am in the middle of a huge amount of work in my current contract, and by the time you read this, I'll hopefully be in a new one.

Errata

In Volume 17 Issue 3 I spotted a couple of errors in the previous instalment. It's funny how you can proofread your own code as often as you like and never spot anything, and then, as soon as you see it in print, errors magically leap out of the page! The problem is on page 45, in the code at label ld_instr.

The comments on exit registers mentions that if D0 is set to -1 on exit, then the required text was not found. This actually means, not found or the input buffer we are looking in, is zero bytes long. In other words, a blank line has been read from the symbol file. The comment should be amended to the following:

```
;       -1 = Not found, or, string being searched is zero bytes long.
```

So, a comment error is not too bad, however, in ld_instr itself, the following two lines exist:

```
        beq.s ldi_done          Unlikely, but you never know!
        bra.s ldi_next          Decrement before comparing
```

These need to be rewritten as follows to ensure that a zero length string results in an exit value in D0, of minus 1.

```
        bne.s ldi_next          Decrement before comparing
        moveq #-1,d0            String length is zero exit code
        bra.s ldi_done          Exit
```

I admit it is unlikely that a symbol file will have any blank lines, but you never know. I have tested the above code changes with a symbol file that has no blanks and another, identical one, to which I added a few blank lines. It works.

So What Now?

Well, I have a half finished application and a few more articles on the Pointer Environment up my sleeve. Time permitting of course. As there will unlikely be a future paper version of QL Today, and I have no idea what the future of a replacement might be, I am setting up a mailing list on my web site so that anyone who wishes to take advantage of the remainder of the series, plus any other work I can think of and have time, to write, can.

You will be required to register on the list with a valid email address and I will also need your name too. My blog gets numerous registrations on a daily basis and most of them are from spam bots hoping to get free spam comments posted on my blog - they don't! Anyone signing up without a valid name and email gets deleted as part of my regular housekeeping. The mailing list will not allow you to register without a name and email address. Please supply your valid name. No nicknames please. That saves me some work clearing out the spam bots as well.

You will not get spammed by me when you register. Traffic will be light I imagine. Whenever I have an article ready, I'll send an email and supply a link where you can obtain the latest article. I'm looking at mailing list software that allows me to add attachments to the emails sent out, but so far, these seem few and far between - at least amongst the ones I'm allowed to use by my hosting company that is.

I know Dave Park mentioned that he would be setting up a Joomla system to replace the printed QL Today, but I haven't heard much for a while, so I'm not sure of progress on that matter.

Anyway, check the web page at **http://qdosmsq.dunbar-it.co.uk** to see if the details of the mailing list have been added, and if so, join up to keep reading the rest of the series. At least with my own mailing list, I'll have a half decent idea of how many readers I actually have! ;-)

The End

So, that's it. I've been writing these articles since the very first volume of QL Today, 17 years ago! It's been a long hard slog at times, and I haven't regretted a minute of it. I'd like to thank my faithful reader(s), George Gwilt who has far better coding skills than I have, and who kept a watchful eye on everything I wrote, offering corrections, bug fixes and observations on just about every article. Thanks George.

Hugh Rooms has commented on my articles as well as offering solutions. And for that I'm grateful.

To all of you who read my articles and never once gave me any feedback, I thank you too. Without you, I wouldn't have as many readers as I have -- but honestly, if you ever get involved in a series like that again, please give the author some feedback -- even just a quick email to say 'hello' or similar. Writing in isolation, for free, is fine, but it's far better to know that your efforts are being read by the 'masses'.

I wish everyone involved in QL Today, best wishes for the future.

Cheers, Norm.

The long lost interview with David Karlin

by Urs König

Before I start

This article was lying around as handwritten notes and document snippets for more than a decade until the upcoming demise of QL Today meant 'now or never'. By pure coincidence another event - of which you will read further down - made it easier for me to complete it. As a kind of farewell present to the loyal readers I have invested some time to write it down in a proper form, get it reviewed and completed with the help of the interviewed person and the editor. Enjoy reading!

Get personal

After my QL development, publishing and trading activities faded out in 1992 and came to an ultimate halt in 1995 I became a silent follower of the scene and my interests in the QL shifted to two major topics 1) people and 2) preservation.

Over the years I have learnt that the QL changed my life in many aspects. Summarized I can say that it changed it in very positive way. I would say that without the QL and its technology I would not have made the career the same way as I finally did.

Rewind

It must have been one day in late 2001 when the virus of investigating the QL's history infected me. About the same time I decided to produce the QL/E distribution CD-ROM with all the QL emulators and a large public domain library and launch it as a kind of birthday present at the QL's 18th launch anniversary on January 12th 2002. My mailshot of that day also reached David Karlin. Just a few days after I had made the first contact with him I had to attend a conference in London. I took this opportunity to meet David in person. Some more emails back and forth and we agreed to meet after work on Friday January 25th 2002 at Highgate station in northern London. I arrived by tube at about 6 p.m. and stepped out into a dark and rainy evening. David was already waiting for me in his golden Lexus in the car park.

Small world

After a short welcome David first wanted to know how I found his email address. To cut a long story short, I was investigating the QL's history for the mailshot at the 18th launch anniversary. I had tried Google-Search which did not show much of relevance. I then emailed a few email addresses I had collected and the best response came from Paul Bass of BSS Audio who wrote me that 1) Yes, he did know the David Karlin I was looking for and 2) David had been MD at BSS Audio, but had left them and was now head of R&D at Sage Computing. Unfortunately he was unable to give me his contact details at

Sage. Some weeks later I stumbled across (using Google-Groups) a newsgroup message (1) of Rupert Goodwins (another former Sinclair employee) where he wrote:

"*Sage.com? Where David Karlin, one of the QL designers, has just gone to be head tech bod?*
Small world... (it is, I lost it under my bed the other night)"

I then made some simple logic and added David's name the same way to ·sage.com as seen in one of the email addresses in that newsgroup message thread[1]. Gotcha!

What followed was a discussion on Google and how amazing this website is. David used Google as his default page/portal at that time. We were also asking ourselves of how they make any money. Our thinking then was that they could make a fortune if they linked their search results with ads or sold the search result ranking. Nowadays we all know what big business selling ads alongside search results has become. Time was flying and we arrived at David's home where his wife Alison and their children Lisa and Sam were already waiting for us.

About the people

David Karlin (born 1958) was the Chief Design Engineer leading the development of the Sinclair QL professional computer at Sinclair Research Ltd in the 1980s. After leaving Sinclair David held several positions in the UK audio electronics and IT industry. Nowadays David runs - together with his wife Alison - his own business, an established website for classical music called Bachtrack[2]; "The world's largest listings and reviews website for classical music" and a new site called One Stop Arts[3]; "The new guide in London's glorious and eclectic arts scene".

Urs König (born 1968) was a teenager when the Sinclair QL was in development and launched to the market. At age of 17 he bought his first personal QL when it was on sale after Christmas 1985. Urs started and ran the COWO Electronic business - supporting all QL compatible systems - until 1992, with some extension to 1995. After many years in the IT industry, Urs nowadays runs an independent IT consulting company called COWO Enterprises LLC.

A nice guy

We had a beer and a chat on god and the world. David likes music, be it acoustic or electronic. There was a guitar hanging on a wall and David proudly expressed the benefits of his DSP controlled sound system. We talked about the QL as well and how it compares or better differs from audio electronic designs (in terms of electronic noise and signal quality). Then it was time for dinner where Alison served a tasty sole fish with potatoes followed by a dessert made of yoghurt and fresh fruits. We talked about family, work and the city. As a former stock trader Alison had a lot to tell in this respect. After the meal she took the children to bed and David and I continued to talk about the QL until at about 11 p.m. when it was time to leave so as not to miss the last tube. David was kind enough to drive me back to the station and I took the tube to return to my hotel. It was a pleasant and enriching evening which I will never forget.

David in his office at Sinclair Research Ltd. in Cambridge in 1983.

A decade later

Over the next decade we stayed in contact in loose form. An email a year, Christmas greetings, a word about how fast our children grow or how the ICT industry is changing or the like was more or less all about it. Around the

[1] - https://groups.google.com/d/msg/comp.sys.sinclair/20wmx-zlcJk/aXX_76XTvlYJ
[2] - http://www.bachtrack.com/
[3] - http://onestoparts.com/

QL's 25th anniversary in 2009 communication increased a bit but then came down to the usual level. While checking the web for QL history updates in early January 2013, I had learned that David visited the "Sommets Musicaux De Gstaad" festival last in 2012 and after an email back and forth it was clear that David he would also attend in 2013. Even our schedules were tight, David had a lot to do at and around the festival. I was to leave for a one week skiing vacation with my family in Les Gets in the French Alps. We agreed to meet on Friday February 8th 2013 in Gstaad for a joint ride on the slopes in the Swiss Alps. As the weather forecast was not that good for Friday - the week already brought a lot of snow - I left home very early at 6:30 a.m. What normally is a 2 hour journey became a 3 hour drive at difficult road conditions. I arrived some 25 minutes late at the hotel and met David in the Lobby just at the moment when he stepped up from the basement with his skis. His wife Alison joined us a few minutes later. Ready to go. Instead of using the free shuttle bus I used my car, but instead of taking the direct route to the desired lower terminus we went an alternative way suggested by the receptionist. This took us around the valley on deep snow covered private roads, we then decided to return to known paths and finally ended up where we wanted to start our skiing. The day started funny. Alison left us to take a 2 hours private ski lesson. We appointed to meet her for lunch at the Snoasis Restaurant on top of Eggli.

David in his home in London in January 2002 holding the QL Today magazine Nov/Dec 2001 (Volume 6, Issue 4) and the QL/E CD-ROM.

Urs and David in February 2013 on the slopes of Gstaad in the Swiss Alps.

A great skiing buddy

David and I went directly up the hill and started what became 2 hours of great skiing on Eggli, Chalberhöni and La Videmanette. The sun played hide and seek with clouds. There were almost no other skiers around and the snow was just awesome. David even tried to ride the deep powder. Skiing may not be the first type of sport you assign to Englishmen, but David is definitely a good skier. While going uphill with ski and chair lifts or even cable cars we had great chats on our lives, family, education, work, IT industry, music, movies (mainly James Bond), the world and guess what?, the QL. On the IT topic, where 10 years ago we talked intensely on Google, this time it was about Rich Internet Applications (RIA) where we discussed our experiences - David developed and maintains the CMS for his businesses in PHP CSS, HTML5) with all sort of technologies - ending up in HTML5 versus Native Apps and how Apple changed the world. At noon we met Alison and her private ski instructor Gilbert in the Snoasis restaurant. We enjoyed

QUANTA

QL USERS AND TINKERERS

Worldwide membership is only by subscription and offers the following benefits:

Bimonthly Magazine — Up to 56 pages
Massive Software Library — All Free!
Free Helpline and Workshops
Free advice on Software and Hardware problems
Regional Subgroups — Is there one near you?

One year membership Subscription £18 (includes eMag) If you want a printed copy of QUANTA magazine, please add the following. For 2013 the rates are:UK 4.00, Europe £22.50 and the RoW £29.50

PayPal Surcharge is about 5% - PayPal (See QUANTA Website)
CASH, Cheques and Postal Orders Accepted.
Email: Membership@quanta.org.uk

Further details from the Membership Secretary
John Gilpin,
181, Urmston Lane, Stretford
Manchester. M32 9EH (UK)
Tel. 0161 865 2872

glühwein, beer, hot chocolate and sausages for lunch and had great conversations in different languages - English, French, German and even Portuguese. David speaks fluent French as he lived a few years in his childhood in Paris. He also speaks Portuguese as he lived a year in Cascais near Lisbon at the Atlantic coast in Portugal. Time was running fast. David and Alison needed to return to the hotel to freshen up and change for the afternoon concert of the festival. Outside the restaurant Alison was so kind and shot a great picture of David and me. A last downhill run then a direct drive back to the hotel, farewell and goodbye. On my way back home, driving was easy as the sun brightened up the afternoon. It was a real pleasure meeting David and Alison again after some many years. We also had great skiing, I really enjoyed it. And talking with them is always a pleasure and enriching.

Questionnaire

You may say yourself, nice story but what about the QL? Well, David was so kind to answer specific questions about Sinclair and the QL.

Before Sinclair

Q: Please tell me a bit about your education and employments before Sinclair.

DK: After studying Engineering and Electrical Sciences at Trinity College, Cambridge, I was hired by Fairchild. First 1 year in the Far East at their production plant, then for 1 ½ years at their R&D site in Palo Alto, CA. While working there I was impressed by the workstations we had. They offered multitasking, windowing, etc. All you could dream about in those times. But those machines were expensive like hell. I thought for myself "I can do this (core unit, CRT, printer) for less than £1000".

Employment at Sinclair

Q: Why/How got you employed by Sinclair?

DK: While working at Fairchild in the US I got married for the first time and as my wife did not like the US I was looking for a job back home in England. I went to see Geoffrey King, a head hunter in Cambridge, about a completely different job, and Geoffrey told me about the job at Sinclair – I was later told that it had been advertised as looking for "the greatest computer designer in the world", although I never actually saw the ad. I met Clive for an interview, we both felt a perfect fit and he offered me my American salary, which was about 30% higher than the going rate in the UK. I immediately decided to take the job.

Q: For which main task(s) were you hired by Sinclair?

DK: To build Sinclair's next computer code-named ZX-83 which eventually became the QL.

Q: From/To which date did you work at Sinclair?

DK: I joined Sinclair in summer 82 (must have been August IIRC) and stayed until the sell-off to Amstrad in April 1986.

Q: What projects have you been involved in, in which order?

DK: It was all about the QL. After the QL was done and teething problems solved, I got bored a bit. Nigel Searle felt that and said to me one time "I don't want to lose you David". I said "The next thing I want to do in my career is to start my own business, so can you give me a commercial role". So I became production manager in one of the 1985 reorganisations. I took over the job from David Chatten, though I believe Dave was still working for Clive, presumably in the C5 arena. In this critical and difficult time at Sinclair we did economics in practice under a difficult cash situation but were able to pay back debts of about £8m until Clive sold the thing off to Amstrad. All but a very few employees were fired then, so it was time for me to move on.

Q: What is your today's relation to the people you had been working with at Sinclair or collaborating partners?

DK: I've met John Mathieson and Richard Miller a few times. I've lost contact with Tony Tebby and many others. I don't see Clive any more. Sadly Ben Cheese who became a close friend died of cancer back in 2001 at the age of 46. Philip Hart, who was manufacturing director at our sub-contractor THORN EMI Datatech became a close friend.

Q: What good/bad/funny memories you have from your time at Sinclair?

DK: My first hour in the office was fairly unforgettable. I showed up at the tiny office in King's Parade, and no-one had a clue who I was or why I was there. So Stuart Honeyball, one of the engineers, said to me "Ah, you've come to help us, have you?" and set me to work opening packets of resistors from Radiospares and putting them into drawers. I clearly remember thinking "Oh, well, if they want to pay me 27 grand a year for cutting up RS packets, who am I to complain?" An hour or so later, Mollie Pearson, Clive's PA, arrived: she *did* know who I was and the job started in earnest. A textbook example of how not to welcome a new employee!

About the QL

Q: Even though there are already several books and articles written by insiders, outsiders or even journalists describing the QL's history and development it never became crystal clear how the project was started, organised and who did have which role/leadership to what extent? Who was the overall project manager? Who was in charge for what sub-projects (electronics, software, production, marketing, etc.)?

DK: There wasn't a single project manager in the conventional sense. I 'did' the hardware and pretty much managed myself, Tony Tebby 'did' the software similarly, Rick Dickinson the industrial design. Jim Westwood was in charge of the engineering lab, so I suppose that theoretically, Tony, Rick and I all reported to Jim; in practice, decisions were taken at meetings chaired by Clive. David Southward contributed the Microdrives. When things went into production, Dave Chatten became in charge of that whole process, assisted by a new project manager called John Munford. John was a pure manufacturing bloke. Another important manufacturing guy for the QL was Chris Cowsley. Nigel Searle was MD of the company and therefore attended many of the meetings and contributed a fair bit to the decision making - and I'm sure made most of the key commercial decisions, with which I wouldn't have been involved at the time - I blush to think of what an ivory tower I lived in.

Q: What tasks (even details) did you do on the QL?

DK: I mainly worked on the electronics, the semi-custom ICs ZX8301 and ZX8302 and the printed circuit board (PCB). I also worked closely with the chip and PCB manufacturers and the production plants. I'm pretty sure that I didn't do any software/firmware. Unlike earlier Sinclair computers I decided to do the custom chips in CMOS VLSI and not in bipolar LSI which I considered already as a dead end technology at the time. Only Ferranti and Fairchild did still push their bipolar designs, the industry trend was already towards CMOS.

Q: What's the most accurate name or description for those chips?

DK: ASIC is the correct generic term: ULA is specifically Ferranti's word for a gate array, a form of ASIC where all the layers except the metal and interconnect are fixed. If I remember rightly, the ZX8301 was a gate array (but I'm not 100% sure anymore), and the ZX8302 was definitely not, which is why it was so painful when there were Microdrive problems – a gate array could be turned round in about a month, while a semi-custom chip like the ZX8302 need six months and some seriously expensive mask charges.

Q: Who else did work on the QL electronics/hardware/production?

DK: Rick Dickinson did the industrial design (case, keyboard) in all aspects (drawings, models, prototypes, injection moulding specifications). The important one for me was Ben Cheese – I'm struggling to remember what the others did on the QL, as opposed to the Spectrum work, which is what John Mathieson and Martin Brennan were mainly engaged in. I'm sure there were lots, but my memory's failing me. I remember Aaron Turner, who was very young when he joined us (even compared to the rest of us!), but I can't actually remember what he was working on.

Q: Was it clear at that time that it will be a professional computer, not a home/hobbyist machine?

DK: In retrospect, I don't think so. I heavily wanted it to be professional; to this day I'm not sure that Clive shared the same vision.

Q: How you personally judge the QL from today's perspective?

DK: Technically it was an excellent product, at least the digital part, not the Microdrives. Compared to the ZX Spectrum which for example had a memory refresh which was out of the specifications. Compare the QL's PCB with the one of the Macintosh and you'll see the higher integration we have reached using the custom chips. In recent years I've often thought about 'what could have been'. You know, the direction of the QL was right (the Macintosh proved that). OK, we had those early teething and supply problems but some of the main reasons we failed in the long run were:

> I don't think Clive really wanted to have a professional computer, more a kind of a more powerful ZX Spectrum. Tony Tebby and I shared the same vision of a real professional computer, a real personal computer. The problem was that we never really resolved this with Clive, so the product ended up falling between two stools.
>
> Clive wanted to make the Microdrives work, but we would have done better to go for floppy disk drives the way Amstrad did. Alan Sugar knew that FDD prices were around GBP

60. He went to Panasonic who had those 3" FDD ready for the market, but were not in a good market position. He said them "You want to enter into the FDD market the big scale? I'll pay you GBP 20 per unit and you're in!". He came back with a deal and his CPC and other machines got working disk drives for astonishingly low cost. Clive decided that we had to go the hard way. We had to invest a tremendous amount of effort, time and money to get the Microdrives working properly. Integrating two analogue devices constructed out of a tape loop, a tape head, a drive motor and an ULA into an apparatus like the QL which is full of noise/RF is a real challenge. Ultimately, we never really made them reliable enough and this turned into the QL's Achilles heel. External Microdrives as used with the ZX Spectrum are a much easier task.

With real FDDs instead of Microdrives and other changes from the Sinclair way of doing it I could have built a professional machine to compete the Macintosh for 1/3 of the price.

Design aspects/flaws

Q: Why the IPC chip was added to the design? In other words: Why the ZX8302 was not covering keyboard/controller ports, sound and RS232 receive on its own? If there was a need to add an additional large chip (microcontroller) like the I8049, why not choose a more advanced chip in the same price range (or below) which could even produce proper sound (e.g. AY-3-8910 as known from many other micros which later in 1985 was being used by Sinclair in the ZX Spectrum 128K)?

DK: We did not succeed to pack all the planned features into one 40 pin chip. I remember for sure that if we had to go for a 48 or more pin package instead of the 40pin we were using it would have cost a tremendous amount more. That would have doubled the cost of the chip, made the PCB more complex, maybe even bigger and more costly. Not sure if the board would still fit into the already designed case. The I8049 was a cheap standard microcontroller and was used as a cost driven compromise. We implemented the needed functionality by on-chip firmware to make it work, at least sort of.

Q: Another issue is why the two RS232 ports did not support separate baud rates? Without this feature the ports could almost never be used to have a printer on one port and a modem on the other working at the same time.

DK: Because in fact, there was only one RS232 receiver, which we tried to multiplex using the RTS and CTS signals - one cost cut too far, which, in retrospect, has to rate as one of the more embarrassing design decisions I've ever made.

Q: The QL's Microdrives use the same ULA as the ZX Spectrum's Microdrives. The way the Microdrives are handled by the QL's ZX8302 chip and by the firmware is different from how it is handled by the ZX Spectrum Interface 1 which was completed and launched several months before the QL in summer 1983. What is the story behind that? What about speed vs capacity vs reliability, etc.?

DK: We couldn't have afforded separate ASICs for the Microdrive logic and everything else the ZX8302 had to do. So it was clear that we were going to use a single ASIC to cover everything. The critical failure was this: there wasn't a way of doing an analogue PLL in the CMOS technology we chose for the ZX8302, so I designed a digital PLL, but a miscommunication between me and Ben Cheese - one of those classic "conversations in corridors" that went unchecked - resulted in me making a wrong assumption about the accuracy of the duty cycle produced by the Ferranti ULA. The Spectrum Microdrives were adequately reliable, if not fantastic, but the inadequacy of the digital PLL was very difficult to fix.

Q: What is the problem with the network hardware/timing/protocol? Were the network ports just a more developed cassette interface with a higher frequency?

DK: Actually, I can't remember it being all that bad - it pales into insignificance in my memory compared with the Microdrive troubles. But it was indeed a single "bit-banging" interface - another of those "we can't add any more logic gates or it will get too expensive" decisions.

Q: What was the original plan for the ZX8301 and how it evolved, got changed?

DK: The ZX8301 is the part of the QL I'm the most proud of. We succeeded to pack all planned functionality into this chip.

Q: Why has a non-common flash option been implemented and not a more desirable and usable 16 colour option (BRIGHT of the 8 base colours or real 16 colours)? Was this because of Teletext/BBS/Videotex/MicroNet/Prestel or the like had flashing characters?

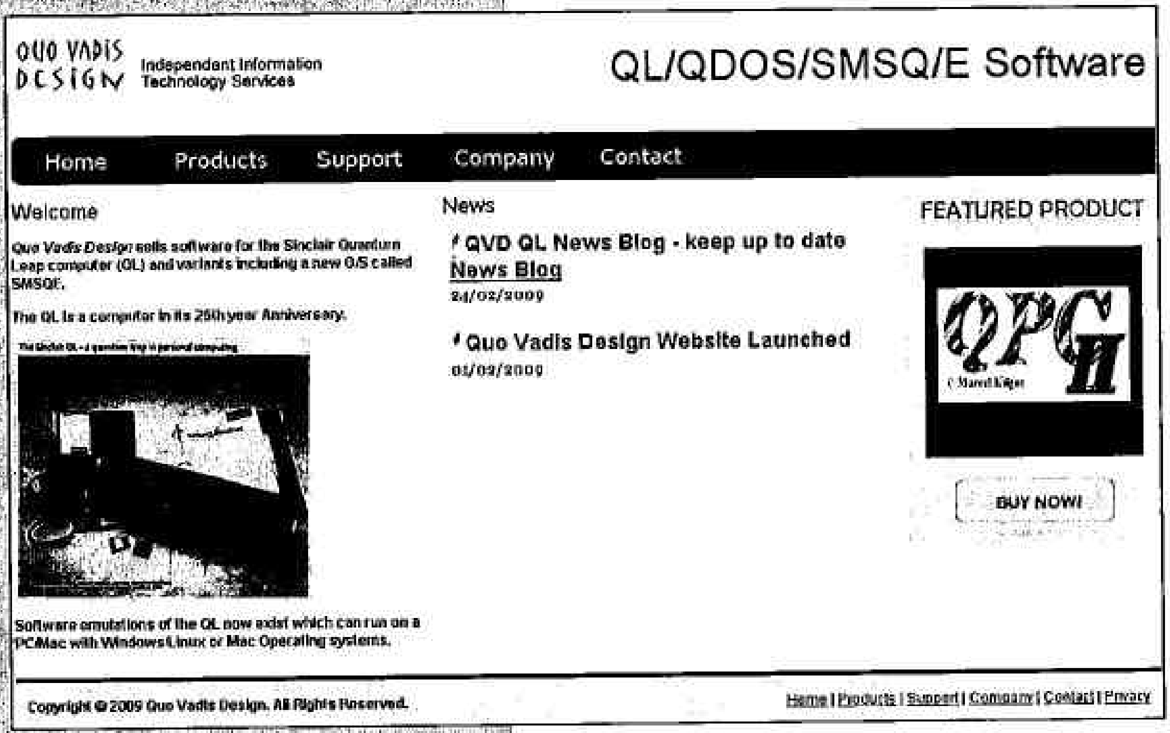
DK: I hardly remember this. The ZX8301 is purely digital, so to get multiple colour levels, we would have needed an extra output which would have

QUO VADIS DESIGN

Independent Information Technology Services

www.ql-qvd.com

QL/QDOS/SMSQ/E Software



QUO VADIS DESIGN Independent Information Technology Services

QL/QDOS/SMSQ/E Software

Welcome

Quo Vadis Design sells software for the Sinclair Quantum Leap computer (QL) and variants including a new O/S called SMSQE.

The QL is a computer in its 25th year Anniversary.

The Sinclair QL - a quantum leap in personal computing

Software emulations of the QL now exist which can run on a PC/Mac with Windows/Linux or Mac Operating systems.

News

- QVD QL News Blog - keep up to date News Blog 24/02/2009
- Quo Vadis Design Website Launched 01/02/2009

FEATURED PRODUCT

Copyright © 2009 Quo Vadis Design. All Rights Reserved.

Home | Products | Support | Company | Contact | Privacy

THANK YOU JOCHEN AND ALL THE QLTODAY EDITORS & CONTRIBUTORS

Bruce@ql-qvd.com

Quo Vadis Design
38 Derham Gardens
Upminster
RM14 3HA
UK

Tel: +44 (0)20 71930539
Fax: +44 (0)870 0568755

Check the QL News Blog on our website for updates.
www.ql-qvd.com/blog

fed a primitive DAC. This would have needed another pin, and we didn't have one.

Q: Reason for adding a TV modulator to a professional machine? Reason the TV output signal was significantly poorer than of most other micros using the same kind of TV modulator. Even late builds the QLs (including Samsung built QLs) produce a very poor quality and blurry TV picture. Remember that the modulator was located close to the Microdrives and power regulation electronics.

DK: The TV modulator came in quite late in the design, maybe two to three months before the launch. It was indeed not following our original plan to produce a professional computer.

Q: Reason for the MC68008 and not the full blown MC68000 (a typical micro of that era runs about 2 to 3 times faster with this chip)? Some said that the chip price was not a big difference if bought in high volumes. Was there an idea of building a cheaper machine because the CPU is 8-bit on the external?

DK: The 68000 was far more expensive than the 68008 at that time (1982/1983). Cost was very important at Sinclair. Saving ½ £ here and there was key.

Q: Why the QL ran at 7.5Mhz (15Mhz quartz) clock rate even though all used standard chips were able to run a 8Mhz? The QL could have 7% more speed at no additional cost. Was there a design decision (Sinclair's flat-screen tube) or timing problem in the ZX8301 at one time during development which made it necessary to lower the system clock to 7.5Mhz?

DK: The design was to work with a CRT monitor not a TV. The DRAM timing was difficult so we used the scanline to the very max. Because you have to match the clock on your CPU to the minimum cycle time of the DRAMs. I can't remember the specifics, but if you get two clock cycles per DRAM access at 8Mhz, you have 250ns in which to cycle your DRAM. If the DRAM won't cycle in that time, you have to add wait states, which is even slower than making the clock faster.

Firmware

Q: What's the real story about the operating system? What was the original plan? Who decided which changes to this plan, when and why?

DK: IIRC, it was decided to run both projects (the in-house and the GST one) in parallel and use whichever came out better. But I really can't remember when that was decided. I assume that Clive and co got frightened about whether Tony and my work was going to deliver, but I don't remember being told this.

Q: What about the initial 32Kbyte ROM design which ended up in 48Kbyte? What was planned to be in the ROM initially (e.g. only the OS, the OS and BASIC, the OS and a cut down BASIC as a kind of Shell/Command Line Interface)? What was planned to be loaded from Microdrive cartridge (e.g. the BASIC)?

DK: I don't think there's any doubt on this one: the ROM was supposed to have OS and BASIC, all apps were supposed to load from the Microdrive.

Q: Who leaded the firmware for the MC68008 (OS, BASIC) and the I8049 (IPC) issues, who (else) was involved in firmware design and writing?

DK: My colleague Tony Tebby did most of the firmware, the operating system. Jan Jones – who was recruited by him – assisted him and wrote the BASIC interpreter.

Manufacturers, production plants, facts and figures

Q: Who manufactured the QL for Sinclair?

DK: THORN EMI Datatech manufactured QLs for us. IIRC we had a dedicated production line at their Feltham plant. I was there quite a few times to discuss details and help them to ramp up production, testing and quality control programmes. They used ROM port dongles as known from early shipped QLs, but those dongles contained various system test software. The production plant in Feltham was managed by Philip Hart.

Samsung became our second source. As UK production was higher than demand we started with them to produce QLs for foreign markets, starting with the US edition and following with the German edition. Samsung made QLs were the better ones, they had much reduced noise as they needed to suite US and German RF regulations. It took us quite a big effort to make the QL US FCC compliant, especially as the product was not designed around such principles and rules. Do you know who knows best such issues? Guess... Yes, professional audio electronic systems as in such systems noise has to be avoided 100%. Based on my experience gained at BSS Audio I could design a noiseless QL nowadays. The PCB layout and lines would look completely different. I visited Samsung in Korea a few times but never saw their production line. IIRC we did not own a dedicated production line for the QL, we got slots to produce batches. You know Samsung was already a giant company back then. Even the hotel which we stayed in was owned by Samsung. Almost "everything" around us in Korea was Samsung. But they were not yet that big in consumer electronics. The ZX Spectrum+ and QL manufacturing was something new for them. It was one of their first sub-contracting jobs. IIRC Dave Chatten found them to do the job for us. Working together was not that complicated. Yes, time was shifted by many hours (UTC+09:00) and language was an issue. No one at Sinclair could speak or understand any Korean, but their English (at least understanding) was good enough to make it work. Remember that at those days we had no email, not even fax yet. Communication was mainly by Telex. We hired a guy (John Munford) at Sinclair who was in charge for the coordination with Samsung. He flew to Korea quite often.

Q: How comes that the QL went through about 18 build-standards and 15 PCB issues during its relatively short production lifespan?

DK: Concerning the high number of build standards, modifications and PCB issues. You know we had those early teething problems which forced us to implement changes. We were under huge pressure because the product was already launched, marketed and sold to customers. Lead time was becoming a problem. A new improved PCB design took about 2 to 3 weeks if not more to get produced and make an impact. Nowadays you have lead times of 1 to 2 days. As there were more and more improvements many were implemented as modifications.

You know production engineering and quality control was a disaster at Sinclair. The reality was somewhere between "severe problem" and "total nightmare". Even some guys like David Chatten were real professionals in pushing sub-contractors and getting cheap prices, Clive's understanding was still the way he experienced in the 60s when he bought factory rejected transistors, measured them, combined those who were working together and built a product out of them, saving some pennies.

I on the other hand came from Fairchild were we already went through a process of improving manufacturing engineering. Of the RAM chips we were producing only about 35% were usable, our Japanese competitors had an output of 80% usable ones. So they could sell for half the price and still making more profit than us. They became a real threat to us and we had to improve our production quality dramatically. I came with this experience to Sinclair.

Q: Were all UK made QLs made by THORN EMI? What about other companies (Timex in Dundee, Scotland or AB Electronics in Abercynon, Wales) who produced products (ZX Spectrums, Spectrum+, ZX Interface 2) for Sinclair? Did they build any QLs or were there at least talks with them to build QLs?

DK: I'm sure AB Electronics did something for us. I'm not sure it was QL related, maybe it was ZX Spectrum related.

QL+

Q: What do you recall of the QL+/Tyche developments? What were the technical specifications for the upgraded QL? How far did the development evolve until the machine stopped?

DK: It was a repacked QL design with additional RAM, minus Microdrives, plus two 3.5" Floppy Disk Drives and a CRT monitor. We did not have the case ready so it was never a real machine.

After Sinclair

Q: How did your career evolve after Sinclair and how did Sinclair and the QL make an impact on your professional life?

DK: After leaving Sinclair in April 1986 I together with Mike Leadbetter, a colleague who also worked for Sinclair, founded a company called Alfa Systems where we developed and marketed a PC network card using only 3 chips making it possible to offer them at a very competitive price. We sold them among others to Sage which I knew from the time at Sinclair where they produced an accounting package which we sold under Sinclair's Silver Label. Alfa went on to market a product called the Diskfax (remember a world in which email attachments didn't really work, and you'll get the idea). From the time working with THORN EMI Datatech on the QL I knew Philip Hart who later also went on and offered me a job in the audio electronics industry, so I became MD at BSS Audio in 1994. At BSS we did 90% export and made about GBP 4Mio revenue per year. When it was again time to move on I got the job at Sage. The two founders of Sage still remembered the QL and my work at Sinclair so the QL helped me to get all new jobs so far. ;-)

David, thank you very much for your time, patience and support.

QL forever!

File Selection

by George Gwilt

Many programs require the user to provide the name of a file.

The obvious way to achieve the result is to ask the user to type in the required name. However, the software package, QMENU, provides another way applicable to programs written for the Pointer Environment. QMENU has sixteen possible menus ranging from selecting files to reporting errors. The menu of interest here is, of course, File Select. In operation an alphabetical list of files in a directory is displayed. One of these can be selected by moving the pointer to it. The range of files can be restricted by specifying an extension, such as _ASM. Selecting an existing file this way is clearly preferable to typing in its name. It is also usually easier to enter a new name by editing a similar one than typing it in full.

There is, however, one restriction on the use of QMENU. Its window must fit inside the host program's window, this being a requirement of PE. The editor QD is one of the programs which employs QMENU to acquire file names. As it happens I use QD frequently and find the restriction slightly annoying. If QD is at or near its minimum size, when I want to load a file it often happens that only a small proportion of the files in the directory are shown. This makes it likely that I will have to use the scroll bars or arrows to locate the file I want to load. To avoid this I usually set QD to near its maximum size so that most, if not all, the directory is shown. The large size of QD's window in turn means that other programs, quite often other QDs, are likely to be hidden. This is a further disadvantage stemming from the restriction mentioned above.

In order to avoid this in my own programs I have in the past resorted to the following expedient. I set the program's window to be resizable and when a file has to be selected I make the window as large as possible before calling QMENU. When the file has been selected, I reset the window to its original size. This, of course, requires the program's window to be resizable.

I will now describe a method which can be used if the window is not resizable. Indeed it can be used even when the window size is too small to accommodate QMENU's File Select. The trick is to call a separate program to select the file. The listing below is for that separate program, FSL v 1.05.

The window definition for FSL has no subsidiary windows or loose items and has a null sprite. The main window is, however, variable from size 512/256 to 1920/1080.

FSL v 1.05

QMENU is an extension Thing called Menus containing several routines including File Select, which has code FSEL. To use FSEL we set up its required parameter list, find the address of FSEL in the QMENU Thing, call it, then free the QMENU Thing.

Parameter List

The first step is to set default entries for FSEL's parameter list. The second step is to read the parameter list set on the stack when the separate program FSL is called. This must have the entry for fs_fnume, where the answer is to be set, and may also have other items as shown in the listing below. Each item is preceded by a minus sign immediately followed by a lower case letter defining the item. Thus -a followed immediately by a long word of value v indicates that v is the address where the answer must be put.

Normal PE Start

Then comes the usual start to a PE program. This is quite routine except that it includes a call to 'get_vec' which sets the Thing vector in A4. This is needed so that a Thing can be first 'used' and then 'freed'. The source code and assembled binary used for get_vec are given as gvec_asm and gvec_bin in gdlibs05_zip on my website gwiltprogs.info. This file also contains the program LEX2_BIN mentioned below.

The second unusual point to note is that instead of a call to read the pointer, the program goes straight to calling for FSEL to select a file. Having done that it stops.

Calling Program

As mentioned above, FSL must be called with a parameter list. To do this I use the program LEX2_BIN whose register usage is:

```
; On entry:-                    On exit:-
;
; A0 -> Prog name               A0 -> Header of JOB
; A1 -> The string for stack
;   (length of string|no of channels|channels|length of str|str)
;   For no channels or pars A1 = 0
; D2.B = Priority               D2 unchanged
; D1.L = Job owner (-1, or 0) D1.L = 0 : D0 is error code from this job
; D3.W = 0 for EX, -1 for EW         =-1 : D0 is return from called job (EW)
;
; ERROR returns in D0 with condition codes set accordingly
```

It is essential that the address of the answer space be set in FSL's parameter list. If the answer is to be set in ANS(A6) the code to set that part of the parameter list might be:

```
          move.b    #'-',6(a1)
          move.b    #'a',7(a1)
          lea       ans(a6),a0
          move.l    a0,8(a1)
```

Source Code for FSL

```
; fsl5_asm
          exp_on                      We need to calculate expressions
          in        win1_lib_hed1     A useful macro for strings

          bra.s     start

          dc.l      0
          dc.w      $4afb
          hed1      <"fsl v1.05">

          in        win1_ass_pe_keys_pe
          in        win1_ass_pe_keys_wman
          in        win1_ass_pe_mc1_keys
          in        win1_ass_pe_keys_menu

sms.fthg  equ       $29    Free a Thing
sms.uthg  equ       $28    Use a Thing

          rsset     0
id        rs.l      1                  Window ID
wmvec     rs.l      1                  WM Vector
slimit    rs.w      4                  size / origin
thvec     rs.l      1                  Thing Vector
parlist   rs.l      15                 Parameter list for FSEL

*
start     lea       (a6,a4.l),a6       dataspace
          bsr       ope                open a con channel . .
          move.l    a0,id(a6)          . . keep the ID
```

```
********************************
* Sets parameter list for FSEL *
********************************

; The only item set here is fs_mennm, the title. Other items
; can be set by the parameter list when fsl5_bin is executed.

; Here the item fs_fname, which points to the answer space,
; has been renamed fs_fnume in my copy of pe_keys_menu. This
; is because the file equ3_sym, used in GWASS, contains a value
; fs_fname as part of the file system channel definition block.
; (See the QL Technical Guide by David Karlin and Tony Tebby)

          lea       parlist(a6),a5
          move.l    #(thp.call+thp.str)<<16,fs_mennm-4(a5)
          lea       title,a1            "Select a File"
          move.l    a1,fs_mennm(a5)     Menu Name

          move.l    #(thp.call+thp.opt+thp.str)<<16,fs_defnm-4(a5)
          clr.l     fs_defnm(a5)        Default Filename

          move.l    #(thp.upd+thp.opt+thp.str)<<16,fs_dirnm-4(a5)
          clr.l     fs_dirnm(a5)        Directory Name

          move.l    #(thp.upd+thp.opt+thp.str)<<16,fs_extnm-4(a5)
          clr.l     fs_extnm(a5)        Extension

          clr.l     fs_lines(a5)        Number of Lines (or 0)

          clr.l     fs_xpos(a5)         x position of left side (or 0)

          clr.l     fs_ypos(a5)         y position of top (or 0)

          move.l    #-1,fs_mainc(a5)    Ignored!!

          move.l    #-1,fs_filec(a5)    Ignored!!

          move.l    #(thp.ret+thp.str)<<16,fs_fnume-4(a5)

                                        fs_fnume -> filename return

*****************************************
* Deal with the incoming parameter list *
* -a -> fs_fnume required - long        *
* -e -> fs_extnm optional - long        *
* -d -> fs_dirnm    "     - long        *
* -n -> fs_defnm    "     - long        *
* -l -> fs_lines    "     - word        *
* -x -> fs_xpos     "     - word        *
* -y -> fs_ypos     "     - word        *
*****************************************

          move.w    (sp)+,d0            no of channels in par list . .
          lsl.w     #2,d0               . . times 4
          adda.w    d0,sp               skip the channels
          move.w    (sp)+,d0            length of par string
          movea.l   sp,a5               A5 -> parameters
          bset      #31,d0              mark not OK pro tem
          move.w    d0,d1               keep length of par string

          bra       st1                 Start decoding the parameters
```

```
st3     subq.w   #1,d0                   adjust count for "-"
        bmi      erexit4   ---->         no marker following "-"!!
        move.b   (a5)+,d2                type of parameter
        lea      poss,a0                 list of markers
        moveq    #possn,d5               number of markers - 1
st7     cmp.b    (a0)+,d2
        beq      st6                     found
        dbf      d5,st7
        bra      st1                     go to next character

; parameter found 0 = y, 1 = x etc

st6     lea      do_tab,a0               list of programs
        add.w    d5,d5
        adda.w   (a0,d5.w),a0
        jmp      (a0)                    do the program

possn   equ      6                       number of options - 1
poss    dc.b     'a','e','d','n','l','x','y'
do_tab  dc.w     do_y-do_tab
        dc.w     do_x-do_tab
        dc.w     do_l-do_tab
        dc.w     do_n-do_tab
        dc.w     do_d-do_tab
        dc.w     do_e-do_tab
        dc.w     do_a-do_tab

do_x    moveq    #fs_xpos,d4
        bra      st8
do_y    moveq    #fs_ypos,d4
        bra      st8
do_l    moveq    #fs_lines,d4

; A word is set

st8     move.w   (a5)+,parlist+2(a6,d4.w)
        subq.w   #2,d0                   adjust count for word
        bra      st9

do_a    bclr     #31,d0                  mark OK
        moveq    #fs_fnume,d4
        bra      st4
do_e    moveq    #fs_extnm,d4
        bra      st4
do_d    moveq    #fs_dirnm,d4
        bra      st4
do_n    moveq    #fs_defnm,d4

; A long word is set

st4     move.l   (a5)+,parlist(a6,d4.w)
        subq.w   #4,d0                   adjust count for long word
st9     bmi      erexit4   ---->         not a proper pointer!!
        bra      st1

st2     move.b   (a5)+,d2                next byte of parameter list
        cmpi.b   #"-",d2                 par marker? . .
        beq      st3                     . . yes
st1     dbf      d0,st2                  get next character
        tst.l    d0                      OK? . .
        bmi      erexit3   ---->         . . no, "-a" has not appeared
```

```
; We have now successfully dealt with the parameters

          adda.w    d1,sp                 clear stack

********************************
* Normal start to a PE program *
********************************

          movea.l   id(a6),a0             set window ID
          moveq     #iop_pinf,d0
          moveq     #-1,d3
          trap      #3
          tst.l     d0                    ptr_gen present? . .
          bne       erexit2    ---->      . . no
          move.l    a1,wmvec(a6)          keep WM vector . .
          beq       erexit2    ---->      . . wasn't there!
          movea.l   a1,a2                 set WM vector in A2
          lea       slimit(a6),a1
          moveq     #0,d2                 this must be zero
          moveq     #iop_flim,d0          maximum size of window . .
          trap      #3
          subi.l    #$C0008,(a1)          . . less 12, 8
          lea       wd0,a3                address of window definition
          suba.l    a4,a4                 clear A4
          moveq     #8,d0
          bsr       get_vec               Find . .
          bmi       erexit1    ---->      . . OOPS! . .
          move.l    a4,thvec(a6)          . . the THING vector
          move.l    #ww0_0,d1             Size of working definition . .
          bsr       getsp                 . . sets ALCHP'd address to A0 . .
          movea.l   a0,a4                 . . and to A4

; We need to set the status area to zeros
; and the loose items if any to "available" (zero)

          lea       wst0,a1               A1 -> status area
          moveq     #wst0_e-wst0-1,d1     bytes to clear - 1
st5       clr.b     (a1,d1.w)
          dbf       d1,st5
          movea.l   id(a6),a0             Replace the channel ID
          move.l    slimit(a6),d1         Maximum window size
          jsr       wm_setup(a2)          Set up the working definition
          moveq     #-1,d1                Set the window . .
          jsr       wm_prpos(a2)          . . where the pointer is
          jsr       wm_wdraw(a2)          Draw the contents

*********************
* Call QMENU's FSEL *
*********************

          bsr       usmen                 A1 points to FSEL . .
          bne       erexit1    ---->

          movea.l   a1,a4                 . . and so does A4
          lea       parlist(a6),a1
          jsr       $18(a4)               Do FSEL

          movem.l   d0/a0/a2,-(sp)
          lea       hmenu,a0
          movea.l   thvec(a6),a4          Free QMENU
          moveq     #sms.fthg,d0
```

The Sinclair QL Forum

A Place for Sinclair QL Users to Meet!

View [illegible] e topics

GENERAL	TOPICS	POSTS	LAST POST
The Welcome Area Introduce yourself here!	23	163	by rgorstellar Fri Feb 10, 2012 3:06 pm
Help for New Users A section for new users to ask for help and also for people to post helpful tips.	14	87	by ellnaa Mon Feb 06, 2012 12:59 pm
General QL Chat A place to discuss general QL issues.	50	336	by [illegible] Fri Feb 10, 2012 3:31 pm
Hardware Bugging hardware related question? Post here!	70	589	by thorsinclair Fri Feb 10, 2012 12:35 pm
S[illegible]re A [illegible] Sof[illegible]	[illegible]	[illegible]	[illegible] 2012 [illegible]
QL Emulation Discussion and advice about emulating the QL on other machines.	22	122	by Brane2 Sun Feb 05, 2012 10:23 pm
Compatibles Corn[illegible] An area to discu[illegible] QL co[illegible]mputers such as the Thor, Auro[illegible]	2	25	by [illegible] Fri Jan [illegible] 11:25 pm

MARKETPLACE	TOPICS	POSTS	LAST POST
For Sale Sell your QL items here!	19	81	by RWAP Thu Jan 19, 2012 5:26 pm

```
         jsr       (a4)
         movem.l   (sp)+,d0/a0/a2

****************
* Exit program *
****************

         bra       sui                 Exit
*
con      dc.w      3
         dc.b      'con'

ope      lea       con,a0              To open "con" . .
         moveq     #-1,d1              . . for this job
         moveq     #0,d3
         moveq     #io_open,d0
         trap      #2
         rts


******************
* Set A1 -> FSEL *
******************

usmen    movem.l   a0/a2/a4,-(sp)
         lea       hmenu,a0
         moveq     #-1,d1
         moveq     #-1,d3
         move.l    #"FSEL",d2
         movea.l   thvec(a6),a4
         moveq     #sms.uthg,d0
         jsr       (a4)
         tst.l     d0
         movem.l   (sp)+,a0/a2/a4
         rts

erexit1  move.w    #-7,([parlist+fs_fnume,a6])
         bra       sui                 exit

erexit2  lea       err2,a5
         bra       erx

erexit3  lea       err3,a5
         bra       erx

erexit4  lea       err4,a5

erx      movea.l   id(a6),a0           set window ID
         moveq     #-1,d1
         moveq     #2,d2
         moveq     #-1,d3
         moveq     #sd_bordr,d0
         trap      #3
         moveq     #2,d1
         moveq     #sd_setpa,d0
         trap      #3
         moveq     #sd_setst,d0
         trap      #3
         moveq     #7,d1
         moveq     #sd_setin,d0
         trap      #3
         moveq    #sd_clear,d0
         trap     #3
         moveq    #3,d1
         moveq    #1,d2
         moveq    #sd_setsz,d0
         trap     #3
         moveq    #8,d1
         moveq    #4,d2
         moveq    #sd_pos,d0
         trap     #3
         movea.l  a5,a1              message
         movea.w  ut_mtext,a2
         jsr      (a2)

susjob   suba.l   a1,a1              set A1 zero
         moveq    #-1,d1             this Job
         move.w   #300,d3            timeout
         moveq    #mt_susjb,d0
         trap     #1
         bra      sui                exit

get_vec  lib      win1_lib_gvec_bin  This sets the THING vector to A4

         hed1     ‹"Select a File"›,title
         hed1     ‹"Menus"›,hmenu
         hed1     ‹"No return address"›,err3
         hed1     ‹" No PE"›,err2
         hed1     ‹"Faulty parameter"›,err4

         in       win1_ass_pe_qmw1_asm         Window definition

         in       win1_ass_pe_peas_sym_lst
         lib      win1_ass_pe_peas_bin

         in       win1_ass_pe_csprc_sym_lst
         lib      win1_ass_pe_csprc_bin
```

Running Servos from QL based Systems

by Ian Burkinshaw

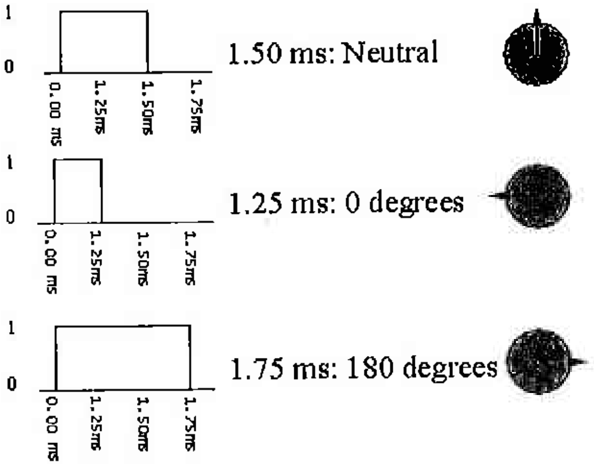
For my final article for QL Today I thought I would build on the theme of interfaces that can be connected to either the USB port on PC based system or the serial (RS232) ports. I have done this with my series on the I2C interface, offering different solutions, and in my review of the PS2 mouse interface in Vol 17 issue 2 page 6 of QL Today which uses the CTL ports. Without having to delve too deeply into the PC or QL hardware.

In this article I shall be looking at driving servos, like the ones used by radio modellers. These can, in fact, be used for all sorts of applications, such as robotics, automation and remote control as well as all sorts of modelling. For example model railway points, which give a far more realistic action to points going over. So not just radio controlled cars and aircraft. In fact I use two of these servos to remotely control a loop antenna, one servo used for rotating the antenna itself and one for tuning the antenna.

What is a servo? Simply a device that converts a control signal into a physical angular position. It can also set a speed for an electric motor. There are motor speed controllers that use the same control signals as positional servos.

In general, analogue servos have three wires, power, ground and signal input. The power requirements of most servos is between 4.5V and 6V. So the power supply for this interface and the servos can be the same. However there are servos around that have different power requirements, for example HV types. These use the same signal as the 5V type servos, but require a higher supply voltage. In the main these HV servos are more powerful. So it is worth checking the power requirements of the servos you wish to use. Also ensure your power supply can supply sufficient current to drive all the servos (motor controllers) at the same time, this can add up.

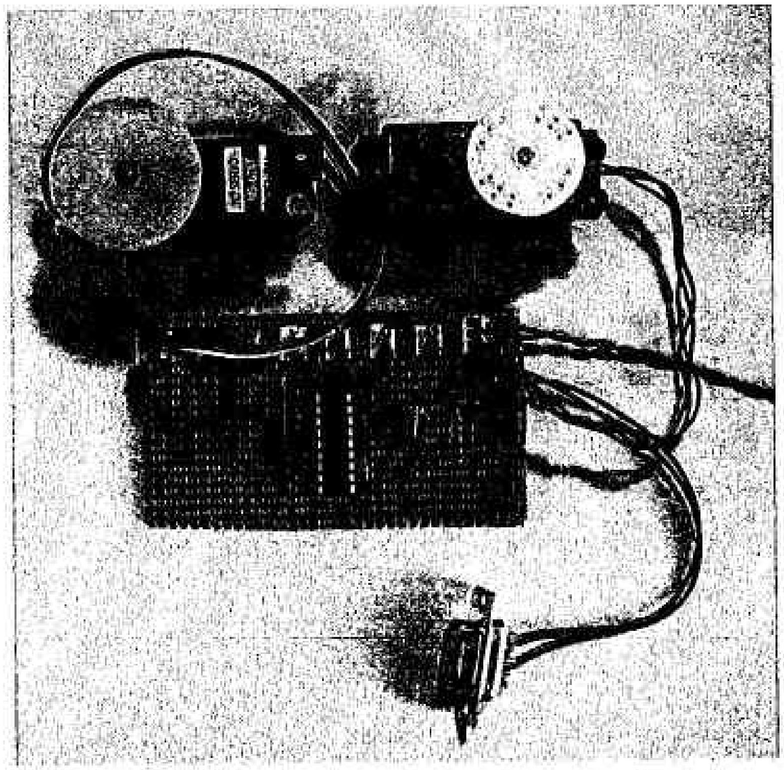
The signal that controls the servo is a pulse sent every 20mSec. It does not have to be this way you can send the pulse just once and the servo will move and stay put. What is important is the width of the pulse, since it is this that sets the angular position of the servo as shown in the right diagram.



Searching the internet I found a PIC project that is ideal for these servos. The interface can drive up to 8 Servos. As it uses the standard RS232 serial interface it can be used with Black Box QL's or with a USB to RS232 serial converter. It can also be used with PC based systems for example QPC2. The interface uses a PIC to translate the command from the computer to the variable pulse width signal required by the servos. The PIC used is PIC16F84, as has been used in some of my other projects. I discussed the topic of programmers in my review of the PS2 Mouse to Games Port Converter in QLT Vol 17, Issue 2, page 6 and in my I2C article in QLT Vol 17, Issue 3, page 48. So if you have made the investment in a programmer, as I have discussed in the past, then you should be OK for this project. Only a minimal number of components are required for this project. A clock crystal , capacitors and resistors. A 5 Volt power supply is required.

I shall not waste space here with the circuit, since it is available from the following web site. Also the required HEX files to program the PIC is are available from this site.

http://www.rentron.com/serialservo.htm



As you can see from my example shown in the left picture, this really is very a simple circuit and lends itself to be constructed on strip (Vero) board.

These are typical connections for some popular servos

Manufacturer	Positive	Negative	Signal
Airtronics (Obsolete)	RED	BLACK (in the middle)	BLACK, WHITE or BLUE
Airtronics / Sanwa (Obsolete)	RED	BLACK	WHITE or YELLOW
Airtronics / Sanwa	RED	BLACK	BLUE or YELLOW
Futaba	RED	BLACK	WHITE
Hitec	RED	BLACK	YELLOW
Japan Radio	RED	BROWN	ORANGE
Tower Hobbies	RED	BLACK	WHITE
Kyosho / Pulsar	RED	BLACK	YELLOW

So having constructed the interface, here is how to use it.

The protocol for the RS232 version is very simple. Just print to the serial port chosen, first the servo CHR$(0 to 7) followed by CHR$(0 to 255) which sets the position of the servo. Note, do not send the line feed (CHR$(10)) or carriage return (CHR$(13)), to stop these, put a semicolon (;) at the end of the print line, as shown below.

Below is a simple test routine to test the interface. This program will first run the servo 10 times from one end of its travel to the other end and then back for each servo in turn. Then it will increment the servo one count at a time. This shows the minimum movement than can be achieved. Again it does this for each servo in turn.

If you find that the servo is being over or under exercised, then adjust the lower_limit and/or upper_limit variables. Not all servo's are the same in the way they respond to a given pulse width.

```
10 BAUD 2400:REMark Baud rate of the PIC16F84 Servo Controller board
20 OPEN#3;ser1i:REMark open serial port one, no handshaking
30 lower_limit=60:REMark value of lower limit of servo, need to be adjust for the
   servo used
40 upper_limit=215:REMark value of upper limit of servo, need to be adjusted for
   the servo used
50 FOR s=0 TO 7:REMark step servos
60 FOR a=1 TO 10:REMark count full deflection test cycles
70 servo_drive s,lower_limit:REMark drive servo to lower limit
80 PAUSE 50:REMark wait
90 servo_drive s,upper_limit:REMark drive servo to upper limit
100 PAUSE 50:REMark wait
110 NEXT a
120 NEXT s
125 REMark second test, to step though the deflection range
130 FOR s=0 TO 7:REMark step servos
140 FOR a=lower_limit TO upper_limit:REMark count servo step's
150 servo_drive s,a:REMark set servo
160 PAUSE 50:REMark wait, needs this time for the data to be transmitted, the PIC
    process the data and the servo itself to repond.
170 NEXT a
180 NEXT s
190 CLOSE#3:REMark close serial port
200 DEFine PROCedure servo_drive(servo,POSITION)
210 PRINT#3;CHR$(servo);CHR$(POSITION);:REMark drive data to servo
220 AT 0,0:PRINT "Servo #";servo+1;" Servo Position ";POSITION;"   ":REMark print
    to screen data set to servo's
230 END DEFine
```

So this is the end for QL Today. Like everybody else I am sorry to see it go. But on the other hand it was inevitable this was going to happen sooner or later. I fully understand Jochen and Geoff's decision. I have been amazed that it has lasted 17 years, which just shows the strength of support from all the editors and contributors over the years, including you the readers, without which there would have been no point. I have enjoyed contributing, and should a replacement come along, in whatever form, I would be happy to continue to contribute. I do hope I have given you some ideas of what can be done in hardware terms with your QL systems. I plan to continue and contribute to Quanta, and I may put all my projects together into some form of book, most likely in PDF form. Since I had started to prepare further articles for QL Today both hardware and software based, they may well still see the light of day. That is assuming there is any interest. So thank you all for reading my articles hope you enjoyed them. Bye for now.

KEY BEEP

by Dilwyn Jones

Some keyboards, and the QL keyboard is one, don't provide a very good feedback when you type. When you've been used to a good keyboard, you gain a confidence that what you type has actually registered.

After using a PC keyboard for a while, going back to a QL keyboard is not quite as reassuring while typing.

On a Sinclair Spectrum, it was possible to add a small beep to each keypress. Given the soft rubbery feel of the original Spectrum keyboard, this was quite an improvement – it provided an improved if unusual form of keyboard feedback while typing. There is also a key click facility available on a QL fitted with a Hermes co-processor.

There is also the risk of finding a few lines into your typing that you had CAPS LOCK on all the time, so you have to go back and correct it.

In this article I'll show how to add a little 'beep' sound to the QL keys, and how to make the beep different depending on whether or not the Caps Lock key is on. Note that while it works on most QL systems, if an emulator does not set the Caps Lock status in the system variable sv_caps, you will only get one keyboard tone, not a different one when Caps Lock is on.

Firstly, I'll show how to achieve something similar using either an SBASIC program or a compiled SuperBASIC job.

```
100 REMark KEYBEEP as an SBASIC or compiled job
110 :
120 REMark default duration and pitch values
130 on_duration%  = 1000 : REMark duration of key beep when CAPS LOCK on
140 off_duration% = 1000 : REMark duration of key beep when CAPS LOCK off
150 on_pitch%     = 1    : REMark key beep value when CAPS LOCK on
160 off_pitch%    = 30   : REMark key beep value when CAPS LOCK off
170 :
180 REMark system variables address
190 svars = 163840 : REMark system variables on QL
200 v$ = VER$
210 IF v$ = 'JSL1' OR v$ = 'HBA' THEN
220   REMark use Minerva or SBASIC system variables function
230   svars = VER$(-2)
240 END IF
250 :
260 REMark the two system variables we need, both word length
270 sv_caps  = svars+136 : REMark Caps Lock system variable
280 sv_arbuf = svars+138 : REMark Auto Repeat buffer
290 :
300 old_key = PEEK_W(sv_arbuf)
310 :
320 REPeat check_keys
330   capslock = PEEK_W(sv_caps)
```

```
340   :
350   new_key = PEEK_W(sv_arbuf)
360   IF new_key <> old_key THEN
370     IF capslock = 0 THEN
380       REMark caps lock off
390       BEEP off_duration%,off_pitch%
400     ELSE
410       REMark capslock on
420       BEEP on_duration%,on_pitch%
430     END IF
440     old_key = new_key
450   END IF
460 END REPeat check_keys
```

This routine just goes round in a loop waiting for a key to be pressed, then plays a sound, the pitch of which depends on whether caps lock is on or not.

The actual values used in the BEEP command are set in lines 130 to 160 of the program. The first pair set the pitch and duration of the note for when Caps Lock is off. The second pair for when it's on.

The actual workings of the program is contained within the loop called check_keys. This reads the required information from two system variables, one of which holds the status of the Caps Lock key, the other reads a system variable called sv_arbuf, which is the auto repeat buffer. This changes value when a new key is pressed, to hold the code of the key pressed. To correctly locate the system variables on an SMSQ/E system we need to check the base address. On an original Sinclair QL this is always address 163840 if the second screen is not in use.

Lines 210 to 240 contain a simple routine which checks if the program is running on a Minerva ROM or SBASIC (Minerva ROMs have VER$='JSL1' while SBASIC systems have VER$='HBA') and if so, it uses the special variant of the VER$ function VER$(-2) to return the base address of the system variables, to ensure the program runs correctly on most QL systems.

Lines 270 to 280 remember the addresses of the two system variables we are interested in.

Line 330 reads whether the Caps Lock key is on or off.

The variable 'old_key' remembers the code of the last key pressed, so that lines 350 and 360 can test if a new key has been pressed. If a new key is pressed, a beep command is issued in line 390 or 420 depending on whether or not Caps Lock is on.

My personal preference is to have a low, short note or click when Caps Lock is off (normal) but a more attention-grabbing higher pitched beep when Caps Lock is on.

If you are using SBASIC on a computer with SMSQ/E installed, this program will run quite nicely in the background as an executed BASIC program with no windows open – it just runs and beeps until you break out of it or remove the job concerned.

If you are using SuperBASIC on a QDOS system, you might be able to compile the program, again with no open windows when it runs. You may like to give the program a fairly low priority so it doesn't hog too much processing power on the system concerned.

It's a program which shows you can achieve quite a lot with just a few lines of BASIC!

One of the program's shortcomings, though, is that it isn't able to detect if you press the same key twice in succession.

So I set about machine coding the program and came up with an extension for BASIC called KEYBEEP which, as its name implies, beeps when you press a key. It uses a slightly more complex form of checking the keyboard, reading the auto repeat delay and count system variables, so that it can beep if the same key is pressed twice. It doesn't beep if you hold a key down (auto repeat), and tries not to interrupt a beep sound already playing on the system.

The code is linked to the scheduler loop list, so that it runs smoothly in the background alongside other programs. This is the first time I have written something to use this list, so the assembler listings shows how to use this linked list as an example, along with how to use the second processor to play sounds and check if sounds are already playing. The code provides an extension called KEYBEEP to turn the sounds off and on and to set the duration and pitch of the sounds if the inbuilt defaults are not to your liking.

This code uses the MT.LSCHD and MT.RSCHD traps to link the routine to the scheduler list and later remove it. In theory at least you could replace the calls to those traps with calls to link the code to the 50/60Hz polled list if you had a good reason to do so, as they are fairly similar. The 50/60Hz polling list is really intended for hardware stuff like reading keyboards or serial ports and mostly runs in supervisor mode, while the scheduler loop tasks run mostly in user mode.

Link a task into the scheduler loop:	`trap #1 with d0 = $1E (MTLSCHD)`
Remove a task from the scheduler loop:	`trap #1 with d0 = $1F (MT.RSCHD)`
Link a task into the polled list:	`trap #1 with d0 = $1C (MT.LPOLL)`
Remove a task from the polled list:	`trap #1 with d0 = $1D (MT.RPOLL)`

In each case, the address of the two long words containing (a) the longword used by QDOS to point to the next routine, and (b) a long word containing the entry address for the code we are adding to the list are at the end of our assembler code. QDOS documentation says that the second long word should be set up before calling the trap – there is little code checking to prevent accidental crashes, so we need to take care to set and check everything up properly before linking in our code.

The routine works on most QL and compatible systems, including QPC2.

KEYBEEP Command Syntax

The KEYBEEP command can take a number of different parameters.

With one parameter:
`KEYBEEP 0`– turns off the keyboard sounds
`KEYBEEP 1`– turns on the keyboard sounds

Without any parameter:
`KEYBEEP`– this toggles the keyboard sounds between off and on.

With two parameters:
`KEYBEEP duration,pitch`– the command lets you vary the sound produced by keys when Caps Lock is off.

With four parameters:
`KEYBEEP off_duration,off_pitch,on_duration,on_pitch`
The first two parameters set the duration and pitch of the sound produced when Caps Lock is off, while the third and fourth parameters set the duration and pitch of sounds produced when Caps Lock is on.

Duration can be a 16-bit word value, while pitch is an 8-bit value from 0 to 255. Note that setting a duration of 0 may cause a sound to play indefinitely, so don't use that. Pitch should be from 1 to 255.

The routines survive a NEW, LOAD or CLEAR command in BASIC. If the sounds get on your nerves after a while, just issue a KEYBEEP 0 command to silence it.

Traps Used

The three main traps used are MT.IPCOM to send commands to the second processor plus MT.LSCHD and MT.RSCHD to handle linking our task to the scheduler loop list.

MT.IPCOM stands for Manager Trap, IPC Communication (IPC is the 8049 second processor in a QL, most emulators and compatibles emulate it, so this code should work on most systems). For further information on how this works, see Andy Pennell's QDOS Companion book, pages 27-31.

Towards the end of the assembler listing, you'll see three blocks of code to send as commands to the IPC. One is for checking if a sound is currently playing (starts at the label 'inpstat'). The other two are for playing the two different sounds.

MT.LSCHD links a new routine into the scheduler loop list. This is a linked list, so the 'link' (two long words) are added to the end of the code. If you'd like more details on how this works and how to set it up, see Andy Pennell's QDOS Companion book, pages 98-100.

MT.RSCHD removes our routine from the linked list when we turn off the key beeps.

The Assembler Listing

The listing was originally produced to assemble with the Computer One assembler, although it should work with most assembler programs.

It's quite well commented, so if you are at all familiar with assembler (you have been reading Norman Dunbar's long running series haven't you?) it should be quite easy to follow.

I hope that QL Today will make this listing available through their website if you'd rather not type it all in. I hope to make it available through my website too, on the Toolkits page:

www.dilwyn.me.uk/tk/index.html

```
* **************************************************
* *    KEYBEEP extension for keyboard click/beep    *
* * Separate beeps for CAPSLOCK on and CAPSLOCK off *
* *      on systems supporting use of .sv_caps.     *
* **************************************************
*
* Action depends on number of parameters in command.
*
* Four parameters:
* KEYBEEP off_duartion,off_pitch,on_duration,on_pitch
*
* Two parameters:
* KEYBEEP off_duration,off_pitch
*
* with one parameter, turns key beep on or off
* KEYBEEP 1      turns on
* KEYBEEP 0      turns off
*
* with no parameters, toggles on/off
* KEYBEEP
*
* N.B. Self-modifying code!
*
* Link into BASIC as extension procedure
*
        lea.l   exts,a1         ;point to list of extensions
        move.w  $110,a2         ;bp.init vector
        jmp     (a2)            ;do it and return to BASIC

* list of extension names and definitions
exts    dc.w    1               ;just 1 extension
        dc.w    keybeep-*       ;procedure is here
        dc.b    7,'KEYBEEP'     ;length of name and name of procedure
        dc.w    0               ;end of procedure definitions
        dc.w    0               ;no functions
        dc.w    0               ;end of function list

* new procedure entry point, action depends on how many parameters
keybeep move.w  $112,a2         ;ca.gtint to fetch parameters
        jsr     (a2)            ;fetch up to 4 parameters
        bne.s   error           ;oops, something went wrong...

        tst.w   d3              ;no parameters?
        beq.s   nopar           ;toggle scheduler loop task on/off

        cmpi.w  #1,d3           ;1 parameter (KEYBEEP 0/1 = off/on)
        beq.s   onepar          ;yes, switch on or off

        cmpi.w  #2,d3           ;2 parameters? KEYBEEP dur1,pitch1
        beq.s   twopar          ;yes, capslock off only

        cmpi.w  #4,d3           ;4 parameters?
        beq.s   fourpar         ;yes, KEYBEEP dur1,pitch1,dur2,pitch2

* error message return for anything else
err_bp  moveq   #-15,d0         ;bad parameter error
error   rts                     ;back to basic with error code in d0
```

```
* KEYBEEP - no parameter, so toggle existing state
nopar   lea     lstlink,a0      ;point to link long words
        tst.l   4(a0)           ;zero (=off)?
        beq.s   turnon          ;currently 0, so turn on key beep
        bra.s   turnoff         ;currently non-zero, so turn off key beep

* KEYBEEP 0 or KEYBEEP 1
* fetch the parameter and check if 0(off) or 1(on)
onepar  tst.w   0(a6,a1.l)      ;0=turn off key beep
        bne.s   turnon          ;1=turn on key beep

* turn off key beep service routine
turnoff lea.l   lstlink,a0      ;address of service routine
        tst.l   4(a0)           ;is it already off?
        beq.s   return1         ;yes it is off, so don't try to remove again!
        moveq   #$1f,d0         ;mt.rschd remove scheduler loop task
        trap    #1
        clr.l   (a0)+           ;zero first long word and point to second
        clr.l   (a0)            ;zero second long word of link area
return1 rts                     ;back to basic

* KEYBEEP off_duration,off_pitch,on_duration,on_pitch
fourpar lea.l   beepon,a3       ;where to store caps on key beep values
        movem.w 4(a6,a1.l),d6-d7 ;pitch and duration for caps on beep
        ror.w   #8,d6           ;duration byte order needs to be reversed
        move.w  d6,10(a3)       ;store duration for caps on beep
        move.b  d7,6(a3)        ;store pitch for caps on beep

* enter here for 2 parameter KEYBEEP off_duration,off_pitch
twopar  lea.l   beepoff,a3      ;where to store caps off key beep values
        movem.w 0(a6,a1.l),d6-d7 ;pitch and duration for caps off beep
        ror.w   #8,d6           ;duration byte order needs to be reversed
        move.w  d6,10(a3)       ;store duration for caps off key beep
        move.b  d7,6(a3)        ;store pitch for caps off key beep

* turn on key beep service routine
turnon  lea.l   lstlink,a0      ;point to the linked list long words
        tst.l   4(a0)           ;is it zero?
        bne.s   return2         ;already on, so no need to link in again
        lea.l   routine,a1      ;point to the service routine
        move.l  a1,4(a0)        ;store our service routine adress in link
        moveq   #$1e,d0         ;mt.lschd scheduler loop task
        trap    #1              ;link our service routine to the list
return2 rts                     ;back to basic

* this is the scheduler loop task itself
routine moveq   #0,d0           ;mt.inf, get system variables address
        trap    #1
        move.w  $8c(a0),d0      ;value of sv.ardel
        subq.w  #1,d0           ;auto repeat delay-1
        cmp.w   $90(a0),d0      ;compare to sv.arcnt
        bne.s   nosound         ;no need to play sound
        subq.w  #1,$90(a0)      ;sv.arcnt=sv.arcnt-1
        lea.l   inpstat,a3      ;pointer to BEEPING IPC command
        moveq   #$11,d0         ;mt.ipcom
        trap    #1              ;check if sound already playing
        btst    #1,d1           ;test BEEPING bit value returned
        bne.s   nosound         ;sound already playing
        tst.w   $88(a0)         ;caps lock off?
        bne.s   capson          ;caps lock is on
capsoff lea.l   beepoff,a3      ;address of sound block for caps off
        bra.s   sound           ;call the mt.ipcom trap
capson  lea.l   beepon,a3       ;address of sound block for caps on
sound   moveq   #$11,d0         ;mt.ipcom
        trap    #1              ;call mt.ipcom to play the sound
nosound rts                     ;service routine exit point

* this is the IPC command to test if sound already playing (like BEEPING)
inpstat dc.b    1               ;1 = report input status
        dc.b    0               ;0 parameters to send
        dc.l    0               ;ignored for this command
        dc.b    2               ;size of reply from IPC-send all bits
        dc.b    0               ;align to even

* this is the IPC command to play the key beep sound for CAPS OFF
beepoff dc.b    10              ;10 = command for sound
        dc.b    8               ;number of parameters to send to IPC
        dc.l    $0000aaaa       ;send all bits of each byte
        dc.b    30              ;pitch 1 value
        dc.b    0               ;pitch 2 value (not used)
        dc.w    0               ;grad_x value (not used)
        dc.w    $8000           ;duration 128 (bytes reverse order)
        dc.b    0               ;wrap+(16*grad_y) (not used)
        dc.b    0               ;fuzziness+(16*randomness)
        dc.b    1               ;no reply values
        dc.b    0               ;align to even

* this is the IPC command to play the key beep sound for CAPS ON
beepon  dc.b    10              ;10 = command for sound
        dc.b    8               ;number of parameters to send to IPC
        dc.l    $0000aaaa       ;send all bits of each byte
        dc.b    1               ;pitch 1 value
        dc.b    0               ;pitch 2 value (not used)
        dc.w    0               ;grad_x value (not used)
        dc.w    $8000           ;duration 128 (bytes reverse order)
        dc.b    0               ;wrap+(16*grad_y) (not used)
        dc.b    0               ;fuzziness+(16*randomness)
        dc.b    1               ;no reply values
        dc.b    0               ;align to even

* acheduler loop task routine linked list entry (2 long words)
lstlink dc.l    0               ;pointer to next link
        dc.l    0               ;address of service routine
```

If you'd rather type in a BASIC program than an assembler listing, enter the one in Figure 3. This will save the extension as a 272 byte file called win1_keybeep_cde.

```
100 REMark Make Keybeep_Cde
110 RESTORE
120 base = RESPR(268)
130 FOR a = base TO base+267
140   READ byte%
150   POKE a,byte%
160 END FOR a
170 SBYTES win1_keybeep_cde,base,268
180 STOP
190 :
200 DATA 67,250,0,8,52,120,1,16
210 DATA 78,210,0,1,0,16,7,75
220 DATA 69,89,66,69,69,80,0,0
230 DATA 0,0,0,0,52,120,1,18
240 DATA 78,146,102,24,74,67,103,22
250 DATA 12,67,0,1,103,28,12,67
260 DATA 0,2,103,68,12,67,0,4
270 DATA 103,42,112,241,78,117,65,250
280 DATA 0,196,74,168,0,4,103,68
290 DATA 96,6,74,118,152,0,102,60
300 DATA 65,250,0,178,74,168,0,4
310 DATA 103,8,112,31,78,65,66,152
320 DATA 66,144,78,117,71,250,0,142
330 DATA 76,182,0,192,152,4,224,94
340 DATA 55,70,0,10,23,71,0,6
350 DATA 71,250,0,106,76,182,0,192
360 DATA 152,0,224,94,55,70,0,10
370 DATA 23,71,0,6,65,250,0,118
380 DATA 74,168,0,4,102,12,67,250
390 DATA 0,12,33,73,0,4,112,30
400 DATA 78,65,78,117,112,0,78,65
410 DATA 48,40,0,140,83,64,176,104
420 DATA 0,144,102,38,83,104,0,144
430 DATA 71,250,0,34,112,17,78,65
440 DATA 8,1,0,1,102,20,74,104
450 DATA 0,136,102,6,71,250,0,22
460 DATA 96,4,71,250,0,32,112,17
470 DATA 78,65,78,117,1,0,0,0
480 DATA 0,0,2,0,10,8,0,0
490 DATA 170,170,30,0,0,0,128,0
500 DATA 0,0,1,0,10,8,0,0
510 DATA 170,170,1,0,0,0,128,0
520 DATA 0,0,1,0,0,0,0,0
530 DATA 0,0,0,0
```

Once assembled, 272 bytes of code are generated. This is loaded as follows, assuming you called it keybeep_cde

```
100 base = RESPR(268)
110 LBYTES win1_keybeep_cde,base
120 CALL base
```

QPC2

Marcel Kilgus has announced the first update to QPC2 in six years:
'It includes the latest SMSQ/E v3.16.
New features:

- Added new keyboard driver that doesn't rely on DirectX anymore. DirectX support in Windows became worse and worse over the years, so now QPC can use the Windows keyboard driver. Much work has been put into this feature to be as compatible as possible, so I hope it works for all countries and that even KEYROW continues to make some sense.
  SMSQ/E keyboard tables like the ones created by Clavier will NOT work with this driver, but the old driver is still in there, just select 'SMSQ/E' as the keyboard driver in the configuration dialog.
- Basic support for floppy disc images. Images can be configured on boot but currently not changed afterwards.
- Various bug fixes including a probably more stable screen driver.
- Some CPU fixes thanks to George Gwilt.'

Marcel adds:
'Don't hesitate because it's called a 'beta', it's very stable, I only want people to test the new keyboard driver first before calling it a release'
http://www.kilgus.net/qpc/downloads.html

[Editor - we hope to have QPC2 Version 4 on the cover DVD for you]

SMSQ/E

Wolfgang Lenerz has announced an upgrade to SMSQ/E to version 3.16:
'It contains Marcel's bug fix for the AURORA COLOUR_NATIVE command.
There is also a new window move mode : move with transparency.

a - New move mode

The WM_MOVEMODE keyword has been extended to accept a new move mode:

```
WM_MOVEMODE 3 : REMark the "full window with transparency" move
```

The window to be moved is made 'transparent' : one can 'see through' it. This is done via 'alpha blending'. Alpha blending requires A LOT of computing power. So, even if your machine can theoretically handle this type of move, in practice it might not be feasible. I don't believe, for example, that the QXL can handle it For Q40/Q60 users, switching on the Cache is advisable...
This type of move is only implemented for display modes where alpha blending actually makes sense, i.e. modes 16, 32 and 33. In other display modes, such as the QL screen modes, or Atari mono modes, this will be redirected to move mode 2.
Please note that you cannot use this move mode with anything but the mouse - the keyboard (cursor keys) will not work.

b - Configuring/setting the move mode

The move modes are configured on a system-wide basis - you cannot have one job moving in mode 0 and the other in mode 1.
Thus, all jobs are affected by the move mode, even those written a long time ago (unless, such as Qlib, the job doesn't use the WMAN move routine).

The move mode can be changed in two ways:
1 - Configure SMSQ/E (WMAN) to a mode of your liking.
2 - Use the new WM_MOVEMODE keyword

This takes one parameter, an integer from 0 to 3:

```
WM_MOVEMODE 0 : REMark the old way
WM_MOVEMODE 1 : REMark the "outline" move
WM_MOVEMODE 2 : REMark the "full window" move
WM_MOVEMODE 3 : REMark the "full window with transparency" move
```

c - Configuring/setting the degree of transparency : WM__MOVEALPHA

You can set how transparent the window is supposed to be when being moved, from nearly totally transparent to totally opaque. This is done by setting the 'alpha value', from 1 (nearly transparent) to 255 (totally opaque).
The alpha value is configured on a system-wide basis - you cannot have one job moving with an alpha value of 100 and the other with 200.
Thus, all jobs are affected by this, even those written a long time ago (unless, such as Qlib, the job doesn't use the WMAN move routine).

The alpha value can be changed in two ways:
1 - Configure SMSQ/E (WMAN) to a value of your liking.
2 - Use the new WM_MOVEALPHA keyword

WM_MOVEALPHA : this new keyword defines the amount of transparency the window should have when moved about, from 1 (nearly transparent) to 255 (totally opaque).

Please note that
1) no check is made on the value passed to this keyword, but only the lower byte is used.
2) a value of 255 is actually equivalent to move mode 2.
3) a value of 0 is allowed but, since this would make the window to be moved totally transparent when it is moved (i.e. you would only ever see the background) this is considered to be an error and a value of 255 will be used!
4) alpha blending requires a lot of computing power - it may be too slow on your machine.'

http://www.wlenerz.com/smsqe/

Wolgang also reported that SMSQmulator has now reached version 1.20.
http://www.wlenerz.com/SMSQmulator

SQRview

Bob Spelten has updated his SQRview program:
'You can now easily browse through your current picture directory when DBAS is installed. SQRview also has its own RLE routines, so sprites can be saved with compression.
QL-PIC files with excess padding, produced by 'other' tools are no longer causing crashes. Also some issues that came up during the testing in SMSQmulator have been fixed. Some in SQRview and some in SMSQ/m. Thanks to Wolfgang.
Also QCoCo and QCP have been updated and tested in SMSQ/m (v1.19).'
http://members.upc.nl/b.spelten/ql/

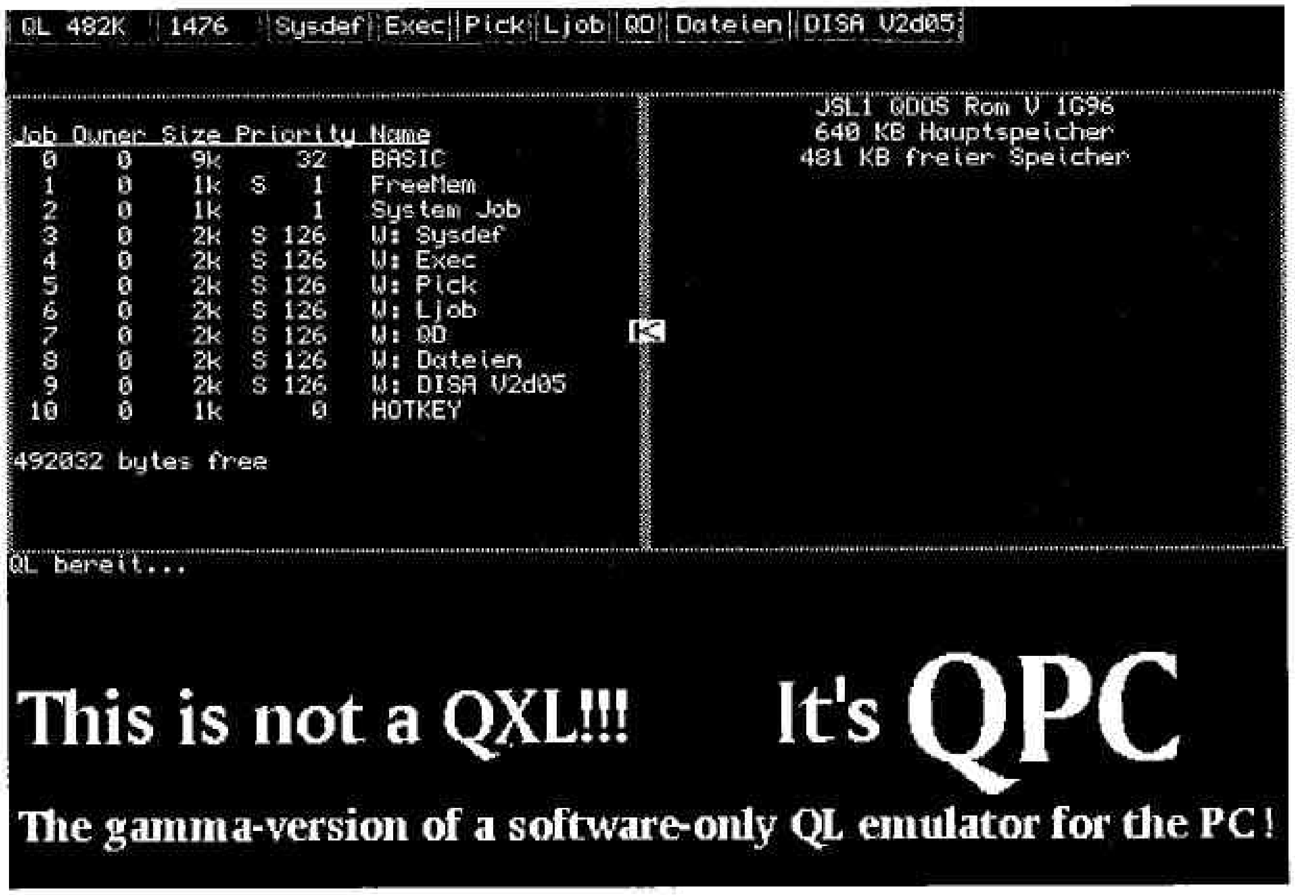
DJToolkit

Dilwyn Jones writes:
'There is an update to the DJToolkit from Norman Dunbar on the Toolkits page of my website.
Version 1.16 fixes a minor issue with the GET_STRING function when used to fetch a null string at the end of a file in SMSQmulator WIN device and RAM device on QPC2.
Download it free from **www.dilwyn.me.uk/tk/index.html**
This issue came to light when we realised Launchpad version 2 wouldn't run from a WIN device on SMSQmulator, so an update to Launchpad will follow later when I've had a chance to do the necessary changes, including a few other very minor updates to it.'

The Story of QPC

by Marcel Kilgus

The story of QPC The year was 1993 when I decided to take on the project of 'PCQL'. I was 13 years old then and have just switched from the QL to a spanking new 486 PC. I did this with some reluctance, but who could resist 16 million colours? Also, most people didn't give the QL scene much more time to live back then. This resulted in some preliminary code to check out if the speed needed for such an undertaking was achievable at all, but nothing more. Then in April 1995 the German club magazine 'Quasar' published its 40th issue which made me realise that the QL was still very much alive, so I revisited the project and renamed it 'QPC', probably to satisfy the scene's 'Q' fetish at the beginning of words. Back then QPC was developed as a real QL replacement which could take on the normal ROMs like JS or MG but worked especially well with Minerva. In September 1995 I showed my work to a few people at a German QL users meeting and while I didn't expect much excitement I was completely floored by the reaction. A few people actually persuaded me to sell them this buggy beta version which couldn't do much more than boot the QL and crash a lot. This of course was a huge deal to a 15 year old boy and spared me from having to deliver papers to earn some cash.



```
QL 482K   1476   Sysdef Exec Pick Ljob QD Dateien DISA V2d05

Job Owner Size Priority Name                    JSL1 QDOS Rom V 1G96
  0    0    9k       32   BASIC                 640 KB Hauptspeicher
  1    0    1k   S    1   FreeMem              481 KB freier Speicher
  2    0    1k        1   System Job
  3    0    2k   S  126   W: Sysdef
  4    0    2k   S  126   W: Exec
  5    0    2k   S  126   W: Pick
  6    0    2k   S  126   W: Ljob
  7    0    2k   S  126   W: QD
  8    0    2k   S  126   W: Dateien
  9    0    2k   S  126   W: DISA V2d05
 10    0    1k        0   HOTKEY

492032 bytes free

QL bereit...
```

This is not a QXL!!! It's QPC

The gamma-version of a software-only QL emulator for the PC!

Back then I apparently though that gamma was less developed than beta

In October my friend Jochen Hassler of DISA and ATR-device fame (who years previously had told me the project was too difficult to pull off, which made me especially persevere) established the contact to Jochen Merz and thus Tony Tebby to see if acquiring the source code to SMSQ/E was an option to develop QPC towards. This resulted in a floppy disc arriving in our mailbox on December the 6th 1995 with large parts of the SMSQ/E source code on it, but there was always something missing and it took until February 1996 until really all files were here.

Of course, never having owned more than a traditional black box QL with some hand-made extensions I didn't own a machine to develop SMSQ/E on, but Jochen Hassler kindly lend me his Super-GoldCard and another friend of mine, Albin Hessler of EasyPtr fame, gave me an old and huge hard drive system with probably 20MB of space. This gear was used all the way until QPC itself was advanced enough to host its own development.

Later in the year 1996 the deed was done and QPC1 was ready for sale. Unfortunately too late to be featured in my beloved Quasar magazine but often written about in this very publication you're holding in your hands right now. It consisted of approximately 15000 lines of x86 assembler code and only ran under a specially booted DOS session. It took over the whole PC and turned it into a QL, just as I had envisioned it. But DOS was on the decline after the release of Windows 95 and soon voices grew loader to have a Windows version. This nagging eventually caused me to buy a few books on Windows and DirectX programming and, to cut along story short, the Windows version QPC2 was released in 1999. It was a pretty straight forward port of QPC1, with new parts written in C (a new language for me back then but my bread and butter to this day) and huge parts still written in x86 assembler. SMSQ/E was being kept compatible with both versions until v2.98, which was solely released for QPC2. Speaking of SMSQ/E, Tony Tebby was of course still actively developing it back then, with me only maintaining the QPC parts and occasionally providing a bug fix now and then.

All in all QPC2 worked exactly like QPC1 before, meaning that it takes over the whole PC while it was active, but at least one could switch to other Windows applications at any time if needed. It took some nagging from a few more people, mainly by my US friend Jim Hunkins while we stayed at his flat near San Francisco, for me to develop QPC2 v2, which could be run in a window right next to other Windows applications. But not only that, it also for the first time sported the new GD2 graphics driver with a QXL compatible 16-bit display mode and the DOS device for access to native Windows files. All this was still a huge processing load for the PCs available back then and was only made workable by porting some parts of the GD2 display driver to PC code.

Further development happened at a breakneck pace and in 2002 QPC2 v2 was pretty soon superseded by QPC2 v3, which until the release of QPC2 v4 in this very magazine remained the current version ever since. It saw huge advances in this timeframe, with BEEP sound support for the NT line of Windows systems (actually based on a disassembly of the original 8049 chip), the SMSQ/E Sample Sound System, an 8-bit display mode (that was mainly developed to debug the Aurora 8-bit driver I was writing at the time), a new 68020 compatible emulation core with the help of George Gwilt, support for the printer emulator QPCPrint, a TCP/IP interface and many more features. Not included in this list were all the things I implemented for all SMSQ/E based platforms after Tony Tebby left the scene, like advanced GD2 drivers with alpha blending, WMAN2, background display I/O, new Ctrl+C switching behaviour, the SBasic command line history and many more.

Huge parts of the current QPC2 codebase are still heavily influenced by code written 15 years ago and if I had to rewrite QPC2 today the whole thing would look a lot different. But considering the huge changes that have happened in this timeframe, from Windows 3.1 to Windows 8, I think it could have been a lot worse. With QPC2 v4 I try to address some of these changes. DirectX as used in QPC for example was the new kid on the block when I first wrote QPC2 but is now supported less and less well with every new Windows and PC generation (emphasis is given to the 3D parts only). Also, Windows applications are not allowed to write in the 'Program Files' directory anymore for security reasons, which means that the scheme of QPC putting its configuration in the SMSQE.BIN file when installed there was no longer working. I tried to address these things in QPC2 v4 to help my loyal users, for example by implementing a new keyboard driver that makes the DirectInput based one obsolete, or re-writing the configuration code to let the SMSQE.BIN settings be overruled by settings in a new QPC.INI file residing in the user directory. Another huge change will be introduced with QPC2 v4: it will no longer be a commercial product. After 17 years on the market it was time to release QPC2 for all to enjoy. QPC started with a resale price of 250 DM (approx. 128 EUR), split between Tony Tebby, Jochen Merz and me. It was reduced over time to the current 60 EUR price tag and now it's finally available for free. During the lifetime of QPC I have finished school, passed through university and finally started a full time job as a software developer. I'm very grateful towards everybody who bought QPC and helped launch my career this way and I look back with fond memories to all the exciting hours of development and meetings with like minded people all over the world. Let's see what the future has in store.

Written in memory of my loving father who was ultimately responsible for introducing me to the QL and computing as a whole and without whom QPC would certainly not exist today.

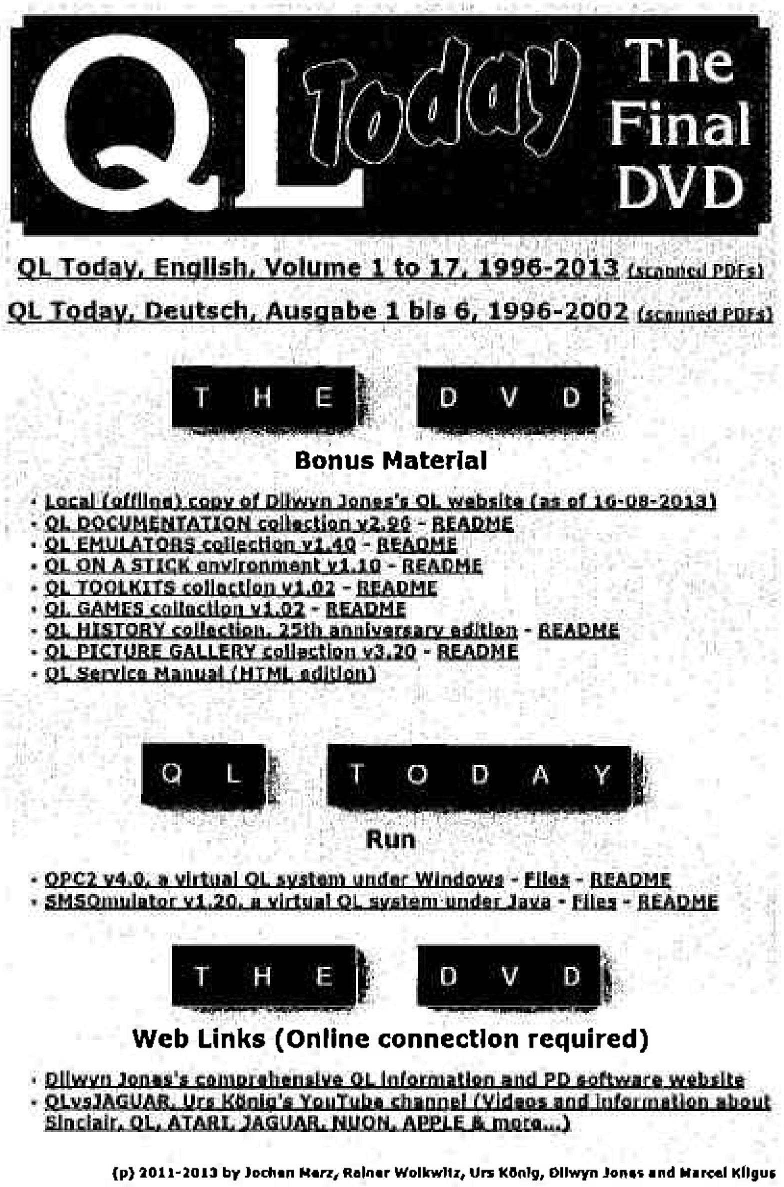
QL Today The Final DVD

by Urs König

The Final DVD is the successor to the DVD which was sent out in September 2011 with Issue 1 of Volume 16 of the QL Today magazine. The new DVD holds everything which was on the old one, but has been carefully updated and supplemented with new material created or preserved in the last two years. The old DVD is obsolete now and can be disposed of or hung on the wall if you like. ;-)

What is it, what it offers?

Please keep in mind that the DVD is designed for use with a computer rather than a DVD player and TV set. So what is the DVD good for? *The Final DVD* holds searchable PDF files of every published QL Today magazine for you to read. Plus an awful lot of bonus material for you to explore. Altogether there are more than 4 GBytes of QL related documents, software and pictures on this DVD. The DVD has a *top level menu* (HTML, file 'index.htm' on root of the DVD) and is laid out in a way that the menu starts automatically on systems configured to allow *auto-start* of DVDs. If it does not auto-start, have a look at the DVD contents using your computer's file manager. Double-click the file 'index.htm' in the top directory to get the menu. The menu system itself is considered to be self-explanatory and presents all the main topics for you to choose from (see screen print).



QL Today, English, Volume 1 to 17, 1996-2013 (scanned PDFs)

QL Today, Deutsch, Ausgabe 1 bis 6, 1996-2002 (scanned PDFs)

Bonus Material

- Local (offline) copy of Dilwyn Jones's QL website (as of 16-08-2013)
- QL DOCUMENTATION collection v2.96 - README
- QL EMULATORS collection v1.40 - README
- QL ON A STICK environment v1.10 - README
- QL TOOLKITS collection v1.02 - README
- QL GAMES collection v1.02 - README
- QL HISTORY collection, 25th anniversary edition - README
- QL PICTURE GALLERY collection v3.20 - README
- QL Service Manual (HTML edition)

Run

- QPC2 v4.0, a virtual QL system under Windows - Files - README
- SMSQmulator v1.20, a virtual QL system under Java - Files - README

Web Links (Online connection required)

- Dilwyn Jones's comprehensive QL information and PD software website
- QLvsJAGUAR, Urs König's YouTube channel (Videos and information about Sinclair, QL, ATARI, JAGUAR, NUON, APPLE & more...)

(p) 2011-2013 by Jochen Merz, Rainer Wolkwitz, Urs König, Dilwyn Jones and Marcel Kilgus

Pic 1. The DVD's top level menu

The top two links lead to an overview of the English and German editions of QL Today respectively. There you can browse through *all the volumes* of the *English* and *German* editions of the *QL Today magazine* and open every single issue. Depending on your computer's PDF reader[1] you can read, navigate, search, copy, mark, print, export or the like.

Next you can browse a local copy of *Dilwyn Jones's comprehensive QL information and PD software website*. You don't need an internet connection for that. Dilwyn's website is known to be the one of the largest and best maintained QL resources on the web.

Then you can dive into every single collection or just open and read the *'README'* (first) document of each *collection*[2]. A HTML edition of the *Sinclair QL Service Manual* - which is much more than a just a pure service manual as it describes the QL's architecture and design - is also available.

[1] http://en.wikipedia.org/wiki/PDF_reader

[2] Collections are what Dilwyn Jones is maintaining under the term CD-ROMs for over a decade now or the QL PICTURE GALLERY produced by the author.

TOPIC	Main type of content	Physical location on DVD	Root QXL.WIN
Scanned PDF files of all QL Today magazines ever published	Documents	\qlt\	–
Local (offline) copy of Dilwyn Jones's QL website (as of 16-08-2013)	Documents, Software, Pictures	\djw\	–
QL DOCUMENTATION collection v2.96	Documents, Software, Pictures	\doc\	X
QL EMULATORS collection v1.40	Software	\emu\	X
QL ON A STICK environment v1.10	Software	\qos\	X
QL TOOLKITS collection v1.02	Software	\tks\	X
QL GAMES collection v1.02	Software	\gms\	X
QL HISTORY collection, 25th anniversary edition	Documents, Software, Pictures	\25a\	–
QL PICTURE GALLERY collection v3.20	Pictures	\pic\	–
QL Service Manual (HTML edition)	Documents, Pictures	\srv\	–

The DVD menu also offers you the possibility to start and use two of the most advanced free QL emulators ***QPC2*** and ***SMSQmulator***. I tend to call them ***virtual QL systems*** as they offer so much more than just emulating a vintage Sinclair QL computer. Both natively run a tailored version of the most advanced QL operating system SMSQ/E in its latest incarnation. If the direct link does not work for you (this may be the case if your computer handles the DVD in a way we were not aware of while preparing it), you can browse their folders using your computer's file manager or read the 'README' (first) documents. Please note that more QL emulators are available in the 'QL EMULATORS' collection and in the 'QL ON A STICK' environment.

Finally the DVD menu offers links to some selected websites which act as a good entry point to the QL World in the World Wide Web. Both websites are unique, offer a large repository and are frequently being updated. More Links to many other QL sites are available on those websites.

Making the most out of your DVD

Occasional users shall be happy with the DVD as it is. Regular users and hardcore QLers will probably copy the DVDs content to their computer(s) or even to their file server(s) and will store the physical DVD away or place it on display in their Retro Computer collection.

Picture galleries easy-peasy

The DVD holds more than 4000 pictures. The vast majority of those are in the two collections 'QL PICTURE GALLERY' and the 'QL HISTORY'. Other places with a relevant quantity of pictures are the 'Local (offline) copy of Dilwyn Jones's QL website' and the 'QL DOCUMENTATION' collection. The easiest way to access/handle the images is to use/browse them with a picture gallery software like Microsoft's 'Windows Essentials[3] Photo Gallery' or Google's 'Picasa[4]'. Both software packages are free downloads at no cost. As long as you've stored the DVD's content to a location known and managed by those software packages (e.g. the 'Documents' folder of your PCs user account) then nothing more needs to be configured. If you like to store the DVDs content in another place (e.g. on your file server), then you must add the network path to your picture gallery software's managed

[3] *http://en.wikipedia.org/wiki/Windows_Essentials*
[4] *http://en.wikipedia.org/wiki/Picasa*

locations. Once this is done, you can use all the features of the software. I personally prefer to use Microsoft's 'Windows Essentials Photo Gallery'. It's easy to use, smart and fast at browsing through large picture collections, good to order pictures and has some nice picture edit features. Some Windows users may prefer using Google's 'Picasa' or another picture manager of their choice. Mac OS X users may use 'iPhoto[5]'.

Usage of the QL software archives

There are a total of thirteen virtual WINchester hard-drives in QXL format (QXL.WIN for short) on the DVD. Five rather large ones, located in the root folders of the QL software collections (see table above), build the core QL software archives of the DVD. Then there's one called 'qpcdemo.win' which is part of the demo version of QPC2 and one called 'SMSQmulator.win' which is part of the SMSQmulator distribution. Both have been updated with QTop-Index, a Benchmark program I wrote way back in the late 1980s. Those two *.WINs are stored twice on the DVD, first in the 'QL EMULATORS' collection and second in the 'QL ON A STICK' environment. This is a case where duplicates can make sense. Then there are four more *.WINs which are stored in ZIP-Files. Two of them hold the 'qpcdemo.win' which was mentioned above. One ('\djw\language\gstqc_qxlwin.zip') holds the complete collection of software written by GST for QDOS. This collection was released into public domain in summer 2010. The last one ('\emu\archives\petawin.zip') is a rather old environment which was prepared way back in 1996 by Peta Jäger for distribution and use with the demo package of QPC version 1.

Starting QPC2 from the DVD's menu will boot the QL environment from 'qpcdemo.win' of the 'QL EMULATORS' collection. Starting QPC2 using 'qpc2.bat' from 'QL ON A STICK' will boot the QL environment from the 'qxl.win' of that environment. Users of QPC2 and SMSQmulator can easily configure them to use specific file system containers in QXL.WIN format for use as WIN1_ to WIN8_. In QPC2 you can even mount such file system containers at runtime using the SBASIC command WIN_DRIVE. WIN_DRIVE even supports relative paths based on where QPC2 has been started of (e.g. WIN_DRIVE 1,'.\..\..\gms\qxl.win'). Users of Q-emuLator can mount QXL.WIN based file system containers on the main screen with a right click to the virtual Microdrive slot. Other QL emulators such as uQLx do also support QXL.WIN.

Using the native filesystem access features of QPC2 (DOSx_), SMSQmulator (NFAx_) and Q-emuLator (Attach Directory) one can easily access more QL software which is stored in ZIP files on various locations on the DVD (e.g. latest version of Turbo BASIC compiler in '\djw\turbo\trbop14.zip').

Thank you

This DVD would not have been possible without the tremendous efforts and work of Jochen Merz, Rainer Wolkwitz, Dilwyn Jones and Marcel Kilgus. Thank you very much guys for making it happen.
This is my last contribution for QL Today but definitely not the last for the QL World. Keep always in mind: 'QL forever'!

The Final Thank You!

by Jochen Merz

Well, strange feeling now, having to write the final few words.
Thanks to everybody who participated right from the start like Stuart Honeyball, Dilwyn Jones, further on like Roy Wood, and reliable until the end to this very final issue with Geoff Wicks and Bruce Nicholls. Thanks to Urs for all the work he put into the first and final DVD. Thanks to all the Authors and Readers... too many to name them all in the remaining space ...
Without you QL Today would have ended many years ago - and without Stuart and Dilwyn, it wouldn't have started. It has been a great pleasure to work with two Editors who both did a great job.
Thanks to all of you!

Jochen

Important - Replacement DVDs

As we learned from the DVD which was shipped with Volume 16 Issue 4, some DVD readers do not like certain types of media. This experience added considerably to the costs last time, and as Volume 17 is way over the budget already, I cannot handle it in the same way.

There is nothing I can do in advance from here ... I burn the DVDs at low speed (the higher the speed, the more problems seem to appear), I use and ship the same type of media, tested, to all readers. If your DVD ROM/DVD writer does not read the DVD, then I suggest you try it on a different computer with a different drive.

If, for any reason, this DVD will not read anywhere, or you would like to have a spare "The Final DVD", then you can order another DVD. I will burn it onto a completely different media without printed cover, which solved most problems two years ago.

However, I cannot replace the free DVD free of charge, nor can I deliver additional DVDs free of charge.

A replacement DVD costs EUR 5,90 (to Germany) or EUR 6,90 (all other countries) incl. shipping. You can order replacement DVDs through the J-M-S website **SMSQ.J-M-S.com** *or send a letter or fax, of course.*

I am sorry I have to do it this way, but the alternative would have been to decide not to add a free DVD to the magazine.

Bye Bye

Tschüß

Adieu

Tot ziens

Adios

Arrivederci

Hwyl fawr